Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Structure & Syllabus

B.Tech. (Computer Engineering)

Effective from Academic Year 2021-22

Prepared by: - Board of Studies in Computer Engineering

Approved by: - Academic Board, Vishwakarma Institute of Technology, Pune

Chairman – BOS **Chairman – Academic Board**

Vision of the Institution

"To be globally acclaimed Institute in Technical Education and Research for holistic Socio-economic development".

Mission of the Institution

- To ensure that 100% students are employable and employed in Industry, Higher Studies, become Entrepreneurs, Civil / Defense Services / Govt. Jobs and other areas like Sports and Theatre.
- To strengthen Academic Practices in terms of Curriculum, Pedagogy, Assessment and Faculty Competence.
- Promote Research Culture among Students and Faculty through Projects and Consultancy.
- To make students Socially Responsible Citizen.

Vision of the Department

"To be a leader in the world of computing education practising creativity and innovation".

Mission of the Department

- To ensure students' employability by developing aptitude, computing, soft, and entrepreneurial skills
- To enhance academic excellence through effective curriculum, blended learning and comprehensive assessment with active participation of industry
- To cultivate research culture resulting in knowledge-base, quality publcations, innovative products and patents
- To develop ethical consciousness among students for social and professional maturity to become responsible citizens

List of Programme Outcomes [PO]

PO	PO Statement
PO1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the

professional engineering practice.

PO7 **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8 **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9 **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10 **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11 **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12 **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

List of PSO Statement

PSO1 Select and incorporate appropriate computing theory principles, data structures and algorithms, programming paradigms to innovatively craft scientific solution addressing complex computing problems.

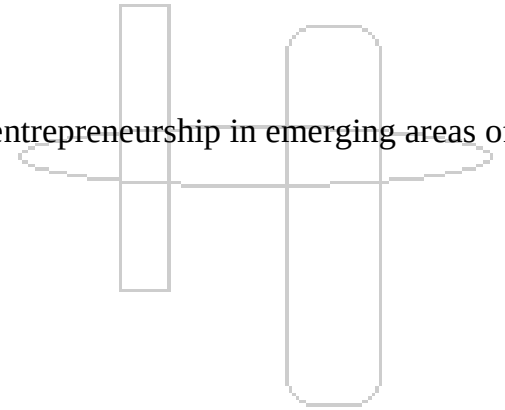
PSO2 Adapt to new frontiers of science, engineering and technology by getting acquainted with heterogeneous computing environments and platforms, computing hardware architectures and organizations through continuous experimentation.

PSO3 Conceive well-formed design specifications and constructs assimilating new design ideas and facts for identified real world problems using relevant development methodologies and practices, architecture styles and design patterns, modeling and simulation, and CASE tools.

PSO4 Exercise research and development aptitude focusing knowledge creation and dissemination through engineering artifacts construction, preparation and presentation of engineering evidences using procedures, techniques, guidelines, and standards considering technology migration and evolution.

Program Educational Objectives (PEOs)

- Demonstrate application of sound engineering foundations to be a committed technology workforce
- Apply mathematical and computing theory knowledge base to provide realistic computer engineering solutions
- Exhibit problem-solving skills and engineering practices to address problems faced by the industry with innovative methods, tools, and techniques
- Develop professional and ethical practices adopting effective guidelines to acquire desired soft skills in the societal and global context
- Aim for continuing education and entrepreneurship in emerging areas of computing



Title: Course Structure **FF No. 653**

Branch: Computer **Year:** S.Y. **A.Y.:** 2021-22 **Module:** III

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Theory	LAB	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)		ESE (%)	PPT/ GD (%)	Viva (%)		
S1	MD2201	Data Science	3	2	1	10	20	10		20	20	20	100	5
S2	CS2221	Internet of Things	3	2	1	10	20	10		20	20	20	100	5
S3	IT2201	Computer Organization and Architecture	3	2	1	10	20	10		20	20	20	100	5
S4	CS2207	Software Development Project-I	0	6	0	-	-	30		70	-	-	100	3
S5	CS2209	Engineering Design and Innovation-III	0	8	0	-	-	30		70	-	-	100	4
S6	CS2218	Object Oriented Programming	3	2	1	10	20	10		20	20	20	100	5
		Total	**12**	**22**	**4**	**40**	**80**	**100**		**220**	**80**	**80**	**600**	**27**

Title: Course Structure **FF No. 653**

Branch: Computer **Year:** S.Y. **A.Y.:** 2021-22 **Module:** IV

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Theory	LAB	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)		ESE (%)	PPT/ GD (%)	Viva (%)		
S1	CS2202	Data Structures	3	2	1	10	20	10		20	20	20	100	5
S2	CS2225	Theory of Computation	3	2	1	10	20	10		20	20	20	100	5
S3	CS2226	Software Engineering	3	2	1	10	20	10		20	20	20	100	5
S4	CS2227	Database Management Systems	3	2	1	10	20	10		20	20	20	100	5
S5	CS2208	Software Development Project-II	0	6	0	-	-	30	-	70	-	-	100	3
S6	CS2210	Engineering Design and Innovation-IV	0	8	0	-	-	30	-	70	-	-	100	4
		Total	**12**	**22**	**4**	**40**	**80**	**100**	**-**	**180**	**80**	**80**	**600**	**27**

Title: Course Structure **FF No. 653**

Branch: Computer **Year:** T.Y. **A.Y.:** 2021-22 **Module: V**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Theory	LAB	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	PPT/ GD (%)	Viva (%)		
S1	CS3205	Design and Analysis of Algorithms	3	2	1	10	20	10		20	20	20	100	5
S2	CS3203	System Software	3	2	1	10	20	10		20	20	20	100	5
S3	CS3215	Web Technology	3	2	1	10	20	10		20	20	20	100	5
S4	CS3217	Database Management Systems	3	2	1	10	20	10		20	20	20	100	5
S5	CS3223	Engineering Design and Innovation-IV	0	8	0	-	-	30	-	70	-	-	100	4
S6	CS3219	Machine Learning	3	2	1	10	20	10		20	20	20	100	5
S6	CS3220	Cyber Security	3	2	1	10	20	10		20	20	20	100	5
		Total	**15**	**20**	**5**	**50**	**100**	**80**		**170**	**100**	**100**	**600**	**29**

Title: Course Structure **FF No. 653**

Branch: Computer **Year:** T.Y. **A.Y.:** 2021-22 **Module: VI**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme							Total	Credits
			Theory	LAB	Tut.	CA		MSA		ESA				
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
S1	CS3216	Theory of Computation	3	2	1	10	20	10		20	20	20	100	5
S2	CS3218	Software Modelling and Design	3	2	1	10	20	10		20	20	20	100	5
S3	CS3202	Artificial Intelligence	3	2	1	10	20	10		20	20	20	100	5
S4	CS3207	Compiler Design	3	2	1	10	20	10		20	20	20	100	5
S5	CS3224	Engineering Design and Innovation-VI	0	8	0	-	-	30	-	70	-	-	100	4
S6	CS3226	Machine Learning Project	3	2	1	10	20	10		20	20	20	100	5
S6	CS3228	Cyber Security Project	3	2	1	10	20	10		20	20	20	100	5
		Total	**15**	**20**	**5**	**50**	**100**	**80**		**170**	**100**	**100**	**600**	**29**

Title: Course Structure **FF No. 653**

Branch: Computer **Year:** BTech **A.Y.:** 2020-21 **Module:** **VII**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Theory	LAB	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
S1	MD4205	Marketing Management	2	0	0	10	-	30	-	30	-	30	100	2
S2	CS4217	Human Computer Interaction	2	0	0	10	-	30	-	30	-	30	100	2
	CS4219	Internet of Things	2	0	0	10	-	30	-	30	-	30	100	2
	CS4222	Image Processing	2	0	0	10	-	30	-	30	-	30	100	2
	ET4230	Natural Language Processing	2	0	0	10	-	30	-	30	-	30	100	2
S3	CS4209	Parallel Computing	2	0	0	10	-	30	-	30	-	30	100	2
	CS4201	Cloud Computing	2	0	0	10	-	30	-	30	-	30	100	2
	ET4232	Deep Learning	2	0	0	10	-	30	-	30	-	30	100	2

S4	CS4225	Major Project	0	20	0		-	30	-	70	-	-	100	10
		Total	**6**	**20**	**0**	**30**	**-**	**120**	**-**	**160**	**-**	**90**	**400**	**16**

Branch: Computer **Year:** BTech **A.Y.:** 2021-22 **Module:** VIII

Subject No.	**Subject Code**	**Subject Name**	**Teaching Scheme (Hrs/Week)**			**Examination Scheme**								**Credits**
			Theory	**LAB**	**Tut.**	**CA**		**MSA**		**ESA**			**Total**	
						HA (%)	**LAB (%)**	**MSE (%)**	**PPT (%)**	**ESE (%)**	**GD (%)**	**Viva (%)**		
S1	CS4233	Industry Internship	0	32	0	-	-	30	-	70	-		100	16
	CS4235	International Internship	0	32	0	-	-	30	-	70	-		100	16
	CS4231	Project Internship	0	32	0	-	-	30	-	70	-		100	16
		Total	**-**	**32**	**-**	**-**	**-**	**30**	**-**	**70**	**-**		**100**	**16**

INDEX

Pattern "B21"

S. Y. B. Tech. Computer Engineering AY 2021-22

Module III Course Content

MD2201:: DATA SCIENCE

Course Prerequisites:

1. Linear Algebra Basics
2. Central Tendency & Measures of Dispersion – Mean, Mode, Median
3. Probability
4. Some exposure to programming environment – C programming; Python

Course Objectives:

1. Understand data processing pipeline
2. Perform dimensionality reduction operations
3. Optimize the performance of functions
4. Apply descriptive statistics tools
5. Deduce meaningful statistical inferences
6. Use unsupervised classification algorithms
7. Use supervised classification algorithms
8. Utilize the data science principles for an entire project life cycle as a case study

Credits: 5

Teaching Scheme Theory: 3 Hours/Week
Tut: 1 Hours/Week
Lab: **2** Hours/Week

Course Relevance:

The course is offered in S.Y. B.Tech. to all branches of Engineering

Data Science is a multidisciplinary field. It uses scientific approaches, procedures, algorithms and frameworks to extract knowledge and insight from a huge amount of data.

Data Science uses concepts and methods which belong to fields like information technology, Mathematics, Statistics, Computer Science etc.

Data Science influences the growth and improvements of the product by providing a lot of intelligence about customers and operations, by using methods such as data mining and data analysis.

The course is relevant to all branches of Engineering and beyond, since data is generated as an obvious outcome of many processes.

SECTION-1

- Introduction to Data Science
  Role of data scientist, introduction to R, R studio; introduction to univariate and multivariate systems, understanding databases, Data Processing - Data collection; Data preparation; Data visualization techniques and inferences - scatter plot, scatter matrix, histogram, box plot.
- Normal distribution, evaluating normal distribution, Binomial distribution, confidence Intervals, central limit Theorem, hypothesis testing, inference for numerical data distribution, paired data, ANOVA
- Vector norms, distances & projections, discriminants, Principal Component Analysis, Optimization: constrained and unconstrained, Gradient Descent

SECTION-2

- Supervised Learning – line fitting, residuals, correlation; line fitting by least squares regression; outliers in linear regression; Inference for linear regression; Multiple regression; Model selection; Logistic regression, Nearest Neighbor Classification – Knn; Naïve Bayes Classification – Bayesian methods, Bayes algorithm; Classification using decision trees and learners
  Unsupervised Clustering - K-means clustering; Evaluation of model performance – Confusion matrices, sensitivity, specificity, kappa statistics, precision, recall, F-measure, ROC curve etc.; Methods of cross-validation, Bootstrapping; Meta-learning through ensemble approach – Bagging, boosting, Random Forests strategies.
  Classifier performance measurement metrics – Training & Testing strategies – Resubstitution, Hold-out, Cross validation, Bootstrap ; Confusion matrix, Performance measures – Accuracy, Error rate, Sensitivity, Specificity, Precision, Recall, F-Measure, Receiver Operating Characteristics curves

List of Tutorials:

1. Data Visualization
2. Distances and Projections
3. Singular Value Decomposition
4. Principal Component Analysis
5. Optimization
6. Normal & Binomial Distribution
7. Hypothesis Testing
8. ANOVA test

9. Linear Regression
10. Logistic Regression
11. Nearest Neighbor Classification
12. Decision Trees based classification
13. Naive Bayes classification
14. Clustering
15. Evaluation of model performance
16. Bagging & Boosting approaches

List of Practicals: (Any Six)

1. Data visualization
2. Unconstrained Optimization
3. Hypothesis Testing
4. Linear regression
5. Logistic Regression
6. Nearest Neighbor classification
7. Naive Bayes classification
8. Clustering
9. Classifier performance using Confusion matrix and other attributes
10. Cross Validation methods

List of Course Projects:

1. Movie recommendation system
2. Customer Segmentation using Machine Learning
3. Sentiment analysis
4. Uber Data analysis
5. Loan prediction
6. HVAC needs forecasting
7. Customer relationship management
8. Clinical decision support systems
9. Development of machine learning solutions using available data sets (multiple projects)
10. Fraud detection

List of Course Seminar Topics:

1. Data wrangling
2. Predictive modeling
3. Data analytics in life science (multiple topics)
4. Ensemble modeling techniques
5. Text pre-processing
6. Feature scaling for machine learning
7. Multivariate normal distribution applications
8. Distance metrics and their applications
9. Visualization techniques such as Chernoff's faces
10. Tree based algorithms
11. Ridge regression
12. LASSO

List of Course Group Discussion Topics:

1. PCA and ICA
2. Hierarchical and nonhierarchical systems
3. Linear - Non linear regression
4. Parametric-non parametric estimation
5. Overfitting and underfitting in the context of classification
6. Linear and Quadratic discriminant analysis
7. Regression v/s classification
8. Classifier performance measures
9. Supervised and unsupervised learning
10. Various clustering approaches
11. Classifiers and classifier combinations
12. Balancing errors in hypothesis testing
13. Standard sampling practices for a successful survey for reliable sample data

List of Home Assignments:

Case Study: A very large number of resources are available for data generated out of case study. Unique Home assignments will be set up for all groups

Surveys: Principles of surveying will be implemented by groups to demonstrate use of data science principles in home assignments

Assessment Scheme:

Mid Semester Examination - 10 Marks
Presentation - 15 Marks
Laboratory - 10 Marks
Course Project - 10 Marks
Home Assignment - 10 Marks
Group Discussion - 15 Marks
End Semester Examination - 10 Marks
Comprehensive Viva Voce - 20 Marks

Text Books: ***(As per IEEE format)***

1. 'A Beginner's Guide to R' – Zuur, Leno, Meesters; Springer, 2009
2. 'Introduction to Data Science' – Igual, Segui; Springer, 2017
3. 'Mathematics for Machine Learning' – Diesenroth, Faisal, Ong; Cambridge University Press, 2017
4. 'Machine Learning with R' – Lantz, Packt Publishing, 2018

Reference Books: ***(As per IEEE format)***

1. 'Elements of Statistical Learning' - Hastie, Tibshirani, Friedman; Springer; 2011
2. 'Data Science from Scratch' - Grus; Google Books; 2015
3. 'The art of Data Science' - Matsui, Peng; 2016
4. 'Machine Learning for absolute beginners' - Theobald; Google Books; 2017

Moocs Links and additional reading material: www.nptelvideos.in

1. https://www.edx.org/course/machine-learning-fundamentals-2
2. https://www.edx.org/course/foundations-of-data-analysis-part-1-statistics-usi
3. https://www.coursera.org/learn/statistical-inference/home/welcome
4. https://www.coursera.org/learn/data-scientists-tools/home/welcome

Course Outcomes:

Upon completion of the course, student will be able to –

1. Apply Data processing & data visualization techniques - 3
2. Implement dimensionality reduction & optimization techniques for enhancing data suitability - 5
3. Perform Descriptive and Inferential statistical analysis for building reliable predictions - 4

4. Implement Supervised algorithms for classification and prediction - 4
5. Implement Unsupervised classification algorithms - 3
6. Evaluate the performance metrics of supervised and unsupervised algorithms - 2
7. Demonstrate complete Data Science life cycle with case studies - 4

CO-PO Mapping

CO1	CO2	CO3	CO4	CO5	CO6
PO1, PO2, PO3,PO6	PO1, PO2, PO3, PO4, PO5, PO12, PSO1	PO1, PO2, PO3, PO4, PO5, PO12, PSO1	PO1, PO2, PO3, PO4, PO5, PSO1, PSO3	PO1, PO2, PO3, PO4, PSO1, PSO3	PO1, PO2, PO3
2, 2, 1, 2	3, 3, 3, 3, 2, 2, 2	3, 3, 3, 3, 2, 2, 2	3,3,3,2,2,3,2	3,3,3,2,3,2	2,2,2

Future Courses Mapping:

1. Deep Learning
2. Reinforcement Learning
3. DBMS
4. Big Data
5. Data Mining
6. Information Retrieval
7. Recommendation Systems
8. Cloud Computing – AWS
9. IOT
10. Artificial Intelligence
11. Pattern Recognition
12. Natural Language Processing
13. Computer Vision
14. Machine Vision
15. Fault Diagnosis
16. Optimization

17. Bioinformatics
18. Computational Biology
19. Econometrics
20. Supply Chain
21. Ergonomics
22. Operations Research
23. Nano-informatics

Job Mapping:

Job opportunities that one can get after learning this course

1. Data Scientist
2. Data Analyst
3. AI Engineer
4. Data Architect.
5. Data Engineer.
6. Statistician.
7. Database Administrator.
8. Business Analyst
9. Business Intelligence Developer
10. Infrastructure Architect
11. Enterprise Architect
12. Machine Learning Engineering
13. Machine Learning Scientist

CS2221::INTERNET OF THINGS

Course Prerequisites:

Students should have a basic Understanding of the Internet, Cloud, Networking Concepts and Sensors

Course Objectives:

The student will be able to

1. Understand IoT Architecture and framework.
2. Recognize and differentiate between the various use cases of different sensors, actuators, solenoid valve etc
3. Learn about fundamental concepts of networking and protocols.
4. Understand IoT Physical, Data link and Higher layer Protocols.
5. Apply theoretical knowledge for Cloud computing.
6. Implement an IoT solution practically

Credits: 5

Teaching Scheme Theory: 3 Hours/Week
Tut: 1 Hours/Week
Lab: **2** Hours/Week

Course Relevance:

The Internet of Things is transforming our physical world into a complex and dynamic system of connected devices on an unprecedented scale. Internet of Things is a system of interrelated computing and sensing devices and has the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

Advances in technology are making possible a more widespread adoption of IoT, from pill-shaped micro-cameras that can pinpoint thousands of images within the body, to smart sensors that can assess crop conditions on a farm, to the smart home devices that are becoming increasingly popular.

IoT is highly relevant in this growing ecosystem of internet-enabled devices. IoT offers increasing opportunities to collect, exchange, analyse and interpret data in real-time. This robust access to data will result in opportunities to further enhance and improve operations. In a world which is moving towards an increasingly connected future, Internet of Things (IoT) is the next big thing. Right from our homes to our cars to our cities, everything is being connected and the technology of IoT is right in the middle of it.

SECTION-1

Introduction to IoT

Physical Design of IOT, Logical Design of IOT, IOT Enabling Technologies, IOT Levels & Deployment Templates, IoT and M2M **(6 Hours)**

IOT Platform Design Methodology

IoT Design Methodology Steps, Home Automation Case Study, Smart Cities, Health Care, Agriculture, Manufacturing and Logistics **(7 Hours)**

IoT Devices

IoT System Design Cycle, Sensors - Terminologies, Calibration, Types, Specification, Use, Actuators - Types and Use, Prototype Development Platform - Arduino / Raspberry pi / Node MCU, Interface with Embedded System **(7 Hours)**

SECTION-1I

Introduction to Wireless Sensor Network

Sensor Node, Smart Sensor Network, Wireless Sensor Network, RFID - Principles and Components, Node MCU **(5 Hours)**

Connectivity Technologies

Network Configuration in IoT, IoT Stack and Web Stack, IEEE 802.15.4 Standard, Zigbee, Bluetooth, Overview of IoT Protocols, MQTT, Cloud Architecture and Types,
Cloud Service Providers **(10 Hours)**

Case Studies (Any Three from following List to be covered☺

Smart lighting, Home Intrusion Detection, Smart Parking, Weather Monitoring System, Weather Report Bot, Air Pollution Monitoring, Forest fire Detection, Smart Irrigation, IoT Printer, IoT in Manufacturing Industry, IoT in Process Industry, IoT in Quality, Control Applications in Industry, IoT in Material Handling System in Industry, IoT in Automobile Industry, Navigation System, Connected Vehicles, Industry 4.0 **(5 Hours)**

List of Practicals: (Minimum Six)

1. Setting up Arduino / Raspberry Pi/ Node MCU ESP8266 : Basic handling , programming
2. LED Interfacing
3. Sensor interface to Node MCU/Arduino / Raspberry Pi Temperature measurement using LM35
4. Actuator interface to Node MCU /Arduino / Raspberry Pi Traffic Signal Control
5. Node MCU /Arduino / Raspberry Pi wireless communication Raspberry Pi as a web server
6. Node MCU/Arduino / Raspberry Pi Cloud interfacing and programming like Thingspeak Email alert using SMTP protocol
7. Sensor data acquisition on Mobile (Mobile APP) / Developing Application (WEB APP) with Django Text transfer using MQTT protocol
8. Home Automation using Cisco Packet Tracer

List of Course Projects:

1. Smart Agriculture System
2. Weather Reporting System
3. Home Automation System
4. Face Recognition Bot
5. Smart Garage Door
6. Smart Alarm Clock
7. Air Pollution Monitoring System
8. Smart Parking System
9. Smart Traffic Management System
10. Smart Cradle System
11. Smart Gas Leakage Detector Bot
12. Streetlight Monitoring System
13. Smart Anti-Theft System
14. Liquid Level Monitoring System
15. Night Patrol Robot
16. Health Monitoring System
17. Smart Irrigation System
18. Flood Detection System
19. Mining Worker Safety Helmet
20. Smart Energy Grid

List of Course Seminar Topics:

1. IoT Architecture
2. Sensor Characteristics
3. IoT for supply chain management and inventory systems
4. IoT Ethics
5. Security in IoT
6. Cloud Computing Platform
7. IoT Best Practices
8. 5G in IoT
9. Middleware Technology
10. M2M energy efficiency routing protocol
11. IoT based Biometric Implementation
12. Complete IoT solution using AWS
13. A smart patient health monitoring system
14. IoT for intelligent traffic monitoring
15. Home automation of lights and fan using IoT

List of Group Discussion Topics:

1. Role of Internet of Things in development of India .
2. Manufacturing industries should make efforts to limit contribution to IoT.
3. Should countries put a ban on IoT for children?
4. Should IoT pay more attention to security rather than just expanding its horizon to the extremes?
5. IoT is the next big thing in technology.
6. IoT poses a huge risk to privacy, if they your system is hacked.
7. IoT is the next big thing for hackers trying to have access to your intimate data.
8. Pros and cons of over-usage of IoT at homes and offices.
9. IoT at battlefields will make life of soldiers safer and easier.
10. IoT will make way for robots to rule over humans one day.
11. IoT devices are making people lazier and obese.
12. IoT needs to be regulated before it goes out of limits and poses serious threat.

List of Home Assignments:

Design:

1. Smart City
2. Smart Transportation
3. Smart Healthcare
4. Smart Industry using IoT
5. Design of IoT framework

Case Study:

1. Open Source in IoT
2. IoT solutions for automobile
3. Cloud Computing
4. AWS
5. Microsoft Azure

Blog:

1. Network Selection for IoT
2. Need of secure protocols
3. Future of IoT
4. IIoT
5. IoT and Industry 4.0

Surveys:

1. Autonomous Vehicles
2. List of Indian companies which offer IoT solutions for agriculture and farming. Describe the problem they are addressing and their solution.
3. Make a list of Indian companies which offer IoT solutions for healthcare. Describe the problem they are addressing and their solution.
4. Make an exhaustive list of everything inside, just outside (immediate surroundings) and on the auto body which must be "observed" for safe and comfortable driving using autonomous vehicles.
5. Compare different Cloud Service providers in the market.

Text Books: ***(As per IEEE format)***

1. Arshdeep Bahga and Vijay Madisetti, "Internet of Things: A Hands-on Approach", (Universities Press)
2. Pethuru Raj and Anupama C. Raman, "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", (CRC Press)

Reference Books:

1. Adrian McEwen, Hakim Cassimally "Designing the Internet of Things", Wiley

2. Ovidiu Vermesan & Peter Friess "Internet of Things Applications - From Research andInnovation to Market Deployment", ISBN:987-87-93102-94-1, River Publishers
3. Joe Biron and Jonathan Follett, "Foundational Elements of an IoT Solution," by Joe Biron

MOOCs Links and additional reading material:

1. https://proed.stanford.edu/course/view.php?id=191
2. https://nptel.ac.in/courses/106/105/106105166/
3. https://create.arduino.cc/projecthub/electropeak/getting-started-w-nodemcu-esp8266-on-arduino-ide-28184f

Course Outcomes

1. Demonstrate fundamental concepts of Internet of Things (CO Attainment level: 2)
2. Recognize IoT Design Methodology Steps (CO Attainment level: 3)
3. Select sensors for different IoT applications (CO Attainment level: 3)
4. Analyze fundamentals of networking (CO Attainment level: 4)
5. Apply basic Protocols in IoT (CO Attainment level: 4)
6. Provide IoT solutions practically with the help of case study (CO Attainment level: 5)

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO3	PO5	PO7	PO9	PO10	PSO3

CO attainment levels

PO3 --3, PO5 ---3, PO7 –2, PO9 ---2, PO10---1, PSO3---3

Future Courses Mapping:

Other courses that can be taken after completion of this course

1. Ad-Hoc Networks

2. Cyber Security
3. Wireless Networks
4. Industry 4.0
5. Big Data

Job Mapping:

The Internet of Things (IoT) is the most emerging field in today's world. It is revolutionizing every industry, from home appliances to agriculture to space exploration. Since the advent of cloud computing, there has been an exponential growth in the number of sensor-enabled devices connected to the internet and expecting further growth accelerating in the coming years. There are diversified career opportunities in this field. The various career positions available as IoT Research Developer, IoT Design Engineer, IoT Product Manager, IoTSoftware Developer, IoT Solution Architect, IoT Service Manager and many more.

Assessment Scheme:

Mid Semester Examination - 10 Marks
Presentation - 15 Marks
Laboratory - 10 Marks
Course Project - 10 Marks
Home Assignment - 10 Marks
Group Discussion - 15 Marks
End Semester Examination - 10 Marks
Comprehensive Viva Voce - 20 Marks

FF No.: 654

CS2218:: OBJECT ORIENTED PROGRAMMING

Course Prerequisites:
Basic course on programming

Course Objectives:

1. Understand Object Oriented programming concepts
2. Demonstrate Object Oriented programming concepts by writing suitable Java programs
3. Model a given computational problem in Object Oriented fashion
4. To develop problem solving ability using Object Oriented programming constructs like multithreading
5. Develop effective solutions using for real world problems using the concepts such as file handling and GUI
6. Implement applications using Java I/O and event-based GUI handling principles
7.

Credits: 5 **Teaching Scheme Theory: 3 Hours/Week**
Tut: 1 Hours/Week
Lab: 2 Hours/Week

Course Relevance:
This is an important course for engineering students. It develops computational problem solving and logic building capability of students. Acquiring programming skills has a high relevance in all branches of Engineering. Once the student gains expertise in coding, this course proves to be beneficial to them to excel in industry demanding coding in specific software.

SECTION-1

Introduction:
What is Object Oriented Programming (OOP)? The need of OOP, Characteristics of OOP.

Java overview: Classes and Objects, Java object storage, Different ways to create objects in Java, Access Modifiers, this reference, main method, Static vs Instance block, Static methods vs Instance methods in Java, Object class, Static class in Java, operators, keywords in java.

Constructors: Constructors in Java, Default constructor, Parameterized constructor, Copy Constructor, Private Constructors and Singleton Classes. **Garbage Collection:** Garbage Collection, How to make object eligible for garbage collection in Java?

Input and Output: Byte Stream vs Character Stream, Command Line arguments, use of Scanner Class, Scanner vs BufferReader Class, Formatted output, Reading input from console.

Arrays in Java: Arrays in Java, initialization, Default Array values, multi dimensional array, passing array to a function, Jagged arrays, java.util.Arrays class, string class, string buffer, string builder.

Methods in Java: Methods, Parameters passing, Returning Multiple values, Throwable fillInStackTrace() method in Java, Valid variants of main(), Variable Arguments (Varargs) method

Inheritance: Inheritance in Java, Types, Constructor in Inheritance, Using final with Inheritance, Accessing superclass member, Override private methods, Parent and Child classes having same data member, Base vs derived class reference. Polymorphism: Method Overloading, Overloading main(), Static vs Dynamic Binding, Method Hiding. Private and final methods, Passing and Returning Objects in Java

SECTION-2

Exception Handling: Exceptions, types, types of handling exception, Checked vs Unchecked Exceptions, Throw and Throws, User-defined Exception, Chained Exceptions.

Interfaces and Abstract Classes: Interface and its usage, Abstract Class and its usage, Difference between Abstract Class and Interface, Nested Interface, Nested Class, Inner class, Anonymous Inner class, Marker interface.

Java Packages: Packages Introduction, default access specifier use, dealing with package.

Collection in Java: Collections Class, Enumeration, Iterators and ListIterator, Using Iterators, Iterator vs Foreach, ArrayList, Vector, Map, Set.

Multithreading: Thread life Cycle, Thread Priority, Thread Methods, Inter-thread Communication, Synchronization, Method and Block Synchronization, Deadlock situation in threading.

File Handling & Database connectivity: File Processing, Primitive Data Processing, Object Data Processing, Wrapper classes, Connecting Java with database (JDBC/ODBC).

Java GUI: AWT, Swing, Components, design patterns. Layout Manager: Flow, Border, Grid and Card. Label, Button, Choice, List, Event Handling (mouse, key), Menus, Tables

List of Course Seminar Topics:

1. Introduction of Arrays and 1D Array programming examples
2. Multidimensional arrays
3. Variants of main() and command line arguments
4. Input and Output stream classes
5. String concepts and various methods of compairing strings
6. Methods in Java
7. Java String Methods
8. Passing array to a function and Jagged array examples
9. Reading input using Scanner and BufferReader Class
10. String, String buffer and String builder
11. Types of Inheritance in Java
12. Implementation of Types using Constructor in Inheritance
13. Using final with Inheritance
14. Base vs derived class reference in Inheritance
15. Using final with Inheritance, Accessing superclass member
16. Parent and Child classes having same data member
17. Overriding, Hiding Fields & Methods
18. Static vs Dynamic Binding & Hiding Methods
19. Private and final methods
20. Passing and Returning Objects in Java
21. Java Memory Management
22. File handling in Java vs C++
23. Data types used in Java vs C++
24. Java Object Serialization and Deserialization

25. Operator precedence
26. Use of Object Class Methods
27. Garbage collection in JAVA
28. Use of Static Blocks in various applications
29. Keywords used in JAVA
30. Types of Variables In JAVA

List of Group Discussion Topics:

1. Checked and unchecked exception, user defined and standard exception
2. Abstraction in Java and different ways to achieve Abstraction
3. Packages in Java – Types, Advantages & Techniques to Access Packages
4. Inner classes, nested interfaces in Java
5. Difference between Interfaces and abstract classes in Java
6. Exception Handling in Java Vs CPP
7. Difference between 1) throw and throws. 2) Final, finally and finalize in Java
8. Discuss Exception propagation and Discuss Exception handling with method overriding in Java
9. Discuss Packages, Access specifiers and Encapsulation in java.
10. Difference between abstraction and encapsulation in Java.
11. Daemon Threads Vs user threads
12. Preemptive scheduling Vs slicing
13. Is it possible to call the run()method directly to start a new thread? pls comment
14. Arraylist Vs Vector
15. Arrays Vs Collections
16. is Iterator a class or an Interface? what is its use?
17. List Vs Set
18. BufferedWriter and BufferedReader classes in java
19. BufferedReader Vs Scanner class in java

20. Buffered Reader Vs FileReader in java
21. Instanceofjava
22. Difference between CPP and JAVA
23. Difference between JDBC and ODBC connectivity
24. file processing in java
25. Difference between premitive data processing and object data processing
26. Creating GUI using swing
27. comparision between Swing, SWT, AWT, SwingX, JGoodies, JavaFX, Apache Pivot
28. Introduction To JFC And GUI Programming In Java
29. Introduction to wrapper classes
30. Why java uses Unicode System?

List of Practicals:

1. Implement Student class using following Concepts
   - All types of Constructors
   - Static variables and instance variables
   - Static blocks and instance blocks
   - Static methods and instance methods

2. There is a class Adder which has two data members of type 1D int array and int variable. It has two functions: getdata and numsum. Function getdata accepts non-empty array of distinct integers from user in 1D int array data member and a targetsum in another data member. The function numsum adds any two elements from an input array which is equal to targetsum and return an array of resulting two elements, in any order. If no two numbers sum up to the target sum, the function should return an empty array. Note that the target sum is to be obtained by summing two different integers in the array; you can't add a single integer to itself in order to obtain the target sum. You can assume that there will be at most one pair of numbers summing up to the target sum. Use constructor. Use extra variables if needed

   Input:
   Array=[3,5,-4,8,11,1,-1,7] targetsum=15
   Output: [8,7]
   Input:
   Array=[3,5,-4,8,11,1,-1,6] targetsum=15
   Output: []

3. Write Java program to calculate area of triangle, square & circle using function overloading. Function parameter accept from user (Use function Overloading concepts and Inheritance).

4. Write a program for following exception, develop a suitable scenario in which the following exceptions occur:
    a. divide by zero
    b. Array index out of bounds exception
    c. Null pointer Exception
5. Write a java program to solve producer-consumer problem where there are two producer threads and one consumer thread.
6. Implement various operations using JDBC Connectivity.
7. Display bank account information (Use interface and inheritance using java)
8. Develop a GUI in java which reads, update the file.

List of Course Projects:

Topics of Course Project would be discussed in Lab session.

List of Home Assignments:

Blog:

1. Single and Multidimensional arrays in Java
2. Comparison Inheritance & Polymorphism
3. Need of abstract classes and interfaces in Java
4. Multithreading concept in Java
5. Signed & Unsigned arithmetic operations usin JAVA
6. Role of start() and run() methods in multithreading

Survey:

1. Strategies for Migration from C++ to Java
2. Product development using Inheritance and Polymorphism in Industry
3. on Java/OOP features popular amongst developers
4. Which other (non-JVM) languages does your application use?
5. How Java Impacted the Internet
6. How can a ArrayList be synchronised without using vector?

Design:

1. Implementation of Singleton design pattern in Java
2. Notes Repository System for Academic
3. Design for employee management system
4. Design for student management system
5. Inventory Management System
6. Write a program to delete duplicate numbers from the file

Case Study:

1. Java development milestones from 1.0 to 16.0
2. Implementation of Different Methods in Polymorphism
3. Real world systems which use java for its implementation
4. Drawing a flag using java
5. Use of different methods of Class object
6. Drawing a flag using java

Assessment Scheme:

Mid Semester Examination - 10 Marks
Presentation - 15 Marks
Laboratory - 10 Marks
Course Project - 10 Marks
Home Assignment - 10 Marks
Group Discussion - 15 Marks
End Semester Examination - 10 Marks
Comprehensive Viva Voce - 20 Marks

Text Books:

Herbert Schildt, "JAVA- The Complete Reference", , 11th Edition, McGraw Hill Education

Reference Books:

1. *Bruce Eckel, "Thinking In Java – The Definitive Introduction to Object-Oriented Programming in the Language of the World-Wide Web", Fourth Edition, Pearson Education, Inc.*

2. R. Morelli and R. Walde, "Java, java, Java – Object-Oriented Problem Solving", 3rd edition, Pearson Education, Inc.

Moocs Links and additional reading material:

Programming using Java| Java Tutorial | By Infosys Technology
https://infyspringboard.onwingspan.com/en/app/toc/lex_auth_01304972186110361645_shared/overview

An Introduction to Programming through C++ – Prof A.G. Ranade- NPTEL- computer science and engineering – NOC https://nptel.ac.in/courses/106/101/106101208/#

Course Outcomes:

The student will be able to –

1. Understand object-oriented programming features
2. Develop real world applications using class, inheritance and polymorphism
3. Adapt Best Practices of Class Design by using Standard Templates Library
4. Solve computing problems by applying the knowledge of Exception handling and Multithreading
5. Design solutions by choosing suitable data structures such as Array, Vector, Map etc
6. Implement applications using Java I/O and event-based GUI handling principles

CO1	CO2	CO3	CO4	CO5	CO6
PO1	PO2, PO3	PO3, PO5, PO7, PSO3	PO4, PO9	PO11	PO12, PSO1, PSO2
3	2, 3	3, 2, 2, 2	3, 1	1	2, 2, 3

Future Courses Mapping:

Advanced Data Structures, Advanced Java, Spring Frame Work, Grails Frame Work

Job Mapping:

Java Programmer, Application Developer, Design Engineer, Senior Software Developer

IT2201:: COMPUTER ORGANIZATION AND ARCHITECTURE

Course Prerequisites:
Basics of computer system and any programming language.

Course Objectives:

1. To study the fundamental concepts of structural Computer system and Computer Arithmetic
2. To understand the basic concepts and functions of Microprocessor
3. To gain knowledge of Computer Memory System
4. To get familiar with GPU and CPU architecture
5. To identify solutions for real world design issues using processors.

Credits: 5

Teaching Scheme Theory: 3 Hours/Week
Tut: 1 Hours/Week
Lab: **2** Hours/Week

Course Relevance:

Modern computer technology requires an understanding of both hardware and software, since the interaction between the two offers a framework for mastering the fundamentals of computing.

The purpose of this course is to cultivate an understanding of modern computing technology through an in-depth study of the interface between hardware and software.

In this course, you will study the history of modern computing technology before learning about modern computer architecture and a number of its essential features, including instruction sets, processor arithmetic and control, the Von Neumann architecture, pipelining, memory management, storage, and other input/output topics.

The course will conclude with a look at the recent switch from sequential processing to parallel processing by looking at the parallel computing models and their programming implications.

SECTION I

Basic concepts of Digital Electronics, Organization and Architecture, Structure & Function, Brief History of computers, Von Neumann Architecture, Integer Representation: Fixed point & Signed numbers. Integer Arithmetic: 2's Complement arithmetic, multiplication, Booth's Algorithm, Division Restoring Algorithm, Non Restoring algorithm, Floating point representation: IEEE Standards for Floating point representations.

8086 Microprocessor Architecture, Register Organization, Instruction types, Types of operands, Instruction formats, addressing modes and address translation. Near & FAR procedure, Instruction cycles. RISC Processors: RISC- Features, CISC Features, Comparison of RISC & CISC Superscalar Processors. Case study of Processor.

Fundamental Concepts: Single Bus CPU organization, Register transfers, Performing an arithmetic/ logic operations, fetching a word from memory, storing a word in memory, Execution of a complete instruction. Micro-operations, Hardwired Control, Example-Multiplier CU. Micro-programmed Control: Microinstructions, Microinstruction-sequencing: Sequencing techniques, Micro-program sequencing

SECTION II

Need, Hierarchical memory system, Characteristics, Size, Access time, Read Cycle time and address space. Main Memory Organization: ROM, RAM, EPROM, E 2 PROM, DRAM, Design examples on DRAM, SDRAM, DDR3, Cache memory Organization: Address mapping. Basic concepts: role of cache memory, Virtual Memory concept. Pipeline and its performance, Data hazards: operand forwarding, handling data hazards in software, side effects. Instruction hazards: unconditional branches, conditional branches and branch prediction.

Parallelism in Uniprocessor system, Evolution of parallel processors, Architectural Classification, Flynn's, Fengs, Handler's Classification, Multiprocessors architecture basics, Parallel Programming Models : Shared memory, Message passing, Performance considerations : Amdahl's law, performance indications.

Parallel computing architectures (multi-core CPUs, GPUs, traditional multi-processor system, Xeon-Phi, Jetson Kit, Kilocore processor), multiprocessor and multicomputer systems, interconnection networks, Modern GPU architecture (in brief), Performance comparison: Speedup, Gain time and scalability.

List of Practical (Any Six)

1. Study of 8086 Architecture and Execution of sample programs.
2. Write 8086 ALP to access marks of 5 subjects stored in array and find overall percentage and display grade according to it.
3. Write 8086 ALP to perform block transfer operation. (Don't use string operations) Data bytes in a block stored in one array transfer to another array. Use debugger to show execution of program.
4. Write 8086 ALP to find and count zeros, positive number and negative number from the array of signed number stored in memory and display magnitude of negative numbers.
5. Write 8086 ALP to convert 4-digit HEX number into equivalent 5-digit BCD number.
6. Write 8086 ALP to convert 5-digit BCD number into equivalent 4-digit HEX number.
7. Write 8086 ALP for following operations on the string entered by the user.
a. String length
b. Reverse of the String
c. Palindrome
8. Write 8086 ALP for following operations on the string entered by the user (Use Extern Far Procedure).
a. Concatenation of two strings
b. Find number of words, lines.
c. Find number of occurrences of substring in the given string.
9. Write 8086 ALP to initialize in graphics mode and display following object on screen.
10. Write 8086 ALP to encrypt and decrypt the given message.
11. Write 8086 ALP to perform following operations on file
a. Open File
b. Write data in the file.
c. Delete data in the file.
d. Close the file.

List of Course Projects:

1. Combinational and Sequential circuits
2. Memory Management
3. Graphics Mode
4. IOT based projects.
5. IoT based atmospheric CO2 administration.
6. IoT based flood risk predictor.
7. Simulate modern traffic control system.
8. Online Parallel Examination.

List of Course Seminar Topics:

1. Computer Architecture VS Computer Organization
2. Evolution of Computing Devices
3. Instructions types , formats and execution
4. Interrupts in Microprocessor
5. Trends in computer architecture
6. RISC Vs CISC architecture : A Case Study
7. ARM processor architecture
8. Latest Technology in Embedded systems
9. Multiplier Control Unit
10. Booth's Encoding Pattern for Fast Scalar Point Multiplication in ECC for Wireless Sensor Networks
11. Internet of Things (IoT) in 5G Wireless Communications
12. State of the art parallel processor design.
13. Memory management in mobile OS.
14. Evolution of processors.
15. Ultra SPARC Processor Architecture.

List of Course Group Discussion Topics:

1. GPU computing: CUDA
2. Memory System
3. Replacement Algorithms
4. Pipelining
5. Cache Coherance
6. Virtural Memory
7. Hazards in pipelining
8. Super Computer

9. Modern computer generations
10. Parallel computing models

List of Home Assignments:

Design:

1. Write the sequence of control steps required for the single bus organization for each of the following instructions:
   1. ADD the (immediate) number NUM to register R1
   2. ADD the contents of memory location NUM to register R1

   Assume that each instruction consists of two words. The first word specifies the operation and addressing mode, and second word contains the number NUM
2. Configure a 32 Mb DRAM chip. Consider cells to be organized in 8K X 4 array. Find out the number of address lines.
3. A set associative cache consists of 64 lines, or slots, divided into four-line sets. Main memory contains 4K blocks of 128 words each. Analyze the format of main memory addresses with proper explanation.
4. A one pipeline system takes 50 ns to process a task. The same task can be processed in 6 segment pipeline with a clock cycle of 10 ns. Determine the speedup ratio of pipeline for 100 tasks. What is maximum speedup ratio?

Case Study:

1. Micro-programmed Control Unit and Hardwired Control Unit.
2. Pipeline Hazards
3. Flynn's architectural classification scheme.
4. Modern Processor units

Survey:

1. New memory technologies and their potential impact on **architecture**
2. Virtual Memory
3. Simulation of a superscalar processor and analyzing impact of design tradeoffs
4. Cache Consistency Models in Modern Microprocessors

Blog:

1. Super Computer
2. Intel Journey
3. New Arm Interconnect technologies
4. Distributed Systems and Parallel Computing

Assessment Scheme:

Mid Semester Examination - 10 Marks
Presentation - 15 Marks
Laboratory - 10 Marks
Course Project - 10 Marks

Home Assignment - 10 Marks
Group Discussion - 15 Marks
End Semester Examination - 10 Marks
Comprehensive Viva Voce - 20 Marks

Text Books:

1. William Stallings, "Computer Organization and Architecture: Designing for Performance", 7th Edition, Pearson Prentice Hall Publication, ISBN 81-7758-9 93-8.
2. C. Hamacher, V. Zvonko, S. Zaky, "Computer Organization", 5th Edition, Tata McGraw Hill Publication, ISBN 007-120411-3.
3. Kai Hwang, "Advanced Computer Architecture", Tata McGraw-Hill ISBN 0-07-113342-9
4. Douglas Hall, "Microprocessors and Interfacing", 2nd Edition, Tata McGraw Hill Publications, ISBN 0-07-025742-6.
5. Peter Abel, "Assembly Language Programming," 5th Edition, Pearson Education Publications, ISBN 10:013030655.

Reference Books:

1. *Hwang and Briggs,"Computer Architecture and Parallel Processing", Tata McGraw Hill Publication ISBN 13: 9780070315563.*
2. A. Tanenbaum, "Structured Computer Organization", Prentice Hall Publication, ISBN 81 –203 – 1553 – 7, 4th Edition.

MOOCs Links and additional reading material:

1. www.nptelvideos.in
2. https://www.udemy.com/
3. https://learn.saylor.org/
4. https://www.coursera.org/
5. https://swayam.gov.in/

Course Outcomes:

Upon completion of the course, post graduates will be able to –

1. Demonstrate computer architecture concepts related to design of modern processors, memories and I/Os. (2)
2. Illustrate the micro operations sequencing. (3)
3. Evaluate various alternatives in processor organization. (3)
4. Understand concepts related to memory & IO organization (2)
5. Adapt the knowledge based on Pipeline and its performance (3)
6. Design real world applications using processors. (4)

Future Courses Mapping:
Advance Computer Architecture, Advance Operating Systems

Job Mapping*:*
Application Developers, System programmer

CS2207::SOFTWARE DEVELOPMENT PROJECT-I

Course Prerequisites: C and Python, Problem Based Learning

Course Objectives:

1. To develop problem solving ability using programming skills by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students

Credits:.3..... **Teaching Scheme Theory:…. Hours/Week**

Tut: … Hours/Week

Lab: 6 Hours/Week

Course Relevance:

Software project development comes under the category of project based learning (PBL).

PBL is an instructional approach designed to give students the opportunity to develop knowledge and skills through engaging projects set around challenges and problems they may face in the real world by using domain specific language technologies.

PBL is "learning by doing."

Modern world sustained and advanced through the successful completion of projects.

In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning.

Project based learning will also redefine the role of teacher as mentor in the learning process.

The PBL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve.

It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy**.**

Teacher's Role in PBL :

- Teacher is not the source of solutions; rather he will act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Selection of Project/Problem:

- The problem-based project oriented model for learning is recommended. The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.
- There are no commonly shared criteria for what constitutes an acceptable project. Projects vary greatly in the depth of the questions explored, the clarity of the learning goals, the content, and structure of the activity.
- Use of technology in meaningful ways to help them investigate, collaborate, analyse, synthesize, and present their learning.
- Activities may include- Solving real life problems, investigation, /study and Writing reports of in depth study, field work.

Student's Role in PBL:

- Students must have the ability to initiate the task/idea .They should not be mere imitators.
- They must learn to think.
- Students working in PBL must be responsible for their own learning.
- . Students must quickly learn how to manage their own learning, Instead of passively

receiving instruction.

- Students in PBL are actively constructing their knowledge and understanding of the situation
- Students in PBL are expected to individually.

Developing Inquiry Skills:

- Students in PBL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution.
- How effective is …….? How strong is the evidence for ……? How clear is ……?
- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.
- Skills required by students in information literacy include:
  How to prepare the search? How to carry out the research
- Sorting and assessing of information in general

Use of Research Methodology: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

Sample Software Project Statement based on Computer Vision with OpenCV

1) Design and deploy a system for traffic board sign detection for moving autonomous car in all-weather conditions.

2) Design and deploy a system for object identification and collision avoidance for unmanned vehicles.

3) Design and deploy a system for real time image compression for interactive real time application.

4) Design and deploy a system for real-time writing Devanagari character detection and conversion in unicode format.

...not limited to.....Faculty and students are free to include other area which meets the society requirements at large.

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

Higher levels of the Booms Taxonomy - analyze, apply, evaluate and create.

Text Books: ***(As per IEEE format)***

1. *A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017*
2. *Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.*
3. *Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robart Capraro, Mary Margaret Capraro*

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from societal need point of view from the core courses

CO2: Recognize algorithms, programming constructs and data structures in building software solutions

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Apply ethical principles in utilizing skills and knowledge to design the reliable and scalable solution to meet challenges using modern tools

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO1	PO2	PO3, PO4	PO5, PO8	PO 6,7,9,12	PO12, PSO1, PSO2
3	3	2, 2	2, 2	2, 2, 2, 1	2, 3, 3

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

Job Mapping:

What are the Job opportunities that one can get after learning this course

Software Engineer, Software Developer, IT Engineer

CS2209::ENGINEERING DESIGN AND INNOVATIONS-III

Course Prerequisites: Problem Based Learning

Course Objectives:

1. To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students

Credits:.1 **Teaching Scheme Theory:…. Hours/Week**

Tut: … Hours/Week

Lab: 8 Hours/Week

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project centric learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Engineering Design" are designed as a ladder to extend connectivity of software technologies to solve real world problems using an interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Group Structure:

- There should be a team/group of 4-5 students.
- A supervisor/mentor teacher assigned to individual groups.

· It is useful to group students of different abilities and nationalities together.

Selection of Project/Problem:

- Students must focus to initiate the task/idea .The idea inception and consideration shall be from following areas as a real world problem:
- Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
- This is the sample list to start with. Faculty and students are free to include other areas which meet the society requirements at large.
- The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher's Role in PCL :

- Teacher is not the source of solutions rather he will they act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Student's Role in PCL:

- Students must have ability to initiate the task/idea .they should not be mere imitators.
- They must learn to think.
- Students working in PCL must be responsible for their own learning.
- Students must quickly learn how to manage their own learning, Instead of passively receiving instruction.
- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution. How effective is …….? How strong is the evidence for ……? How clear is ……?
- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning

- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.
- Skills required by students in information literacy include:
- · How to prepare the search? How to carry out the research
- Sorting and assessing of information in general

Use of Research Methodology: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

ED Sample Case Studies : -

With the adaptation of industry communication standards, Raspberry Pi and Sensors projects can be taken up :

1) Design of wireless voice controlled fire extinguisher in societies and organizations.

2) Design of wireless energy meter reading and sending meter readings to MSEB office.

3) Design of remote controlled automatic light on-off systems in societies and organizations.

4) Design of RFID based library management system.

5) Design of wireless fingerprint based college attendance system.

6) Design of wireless home automation system.

...not limited to.....Faculty and students are free to include other area which meets the society requirements at large.

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

Higher levels of the Booms Taxonomy - analyze, apply, evaluate and create.

Text Books: ***(As per IEEE format)***

1. *A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017*
2. *Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.*
1. *Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robart Capraro, Mary Margaret Capraro*

Reference Books: ***(As per IEEE format)***

1. *De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based learning in engineering. Rotterdam: Sense Publishers. 2007.*
2. *Project management core textbook, second edition, Indian Edition , by Gopalan.*
3. *The Art of Agile Development. By James Shore & Shane Warden.*

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from societal need point of view

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO1	PO2	PO3	PO4, PO5, PO8	PO6, 7, 9, 11	PO12, PSO1, PSO2, PSO4
3	3	2	2, 2, 2	2, 2, 2, 1	2, 3, 2, 2

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

Future Courses Mapping:

Mention other courses that can be taken after completion of this course

Job Mapping:

What are the Job opportunities that one can get after learning this course

Software Engineer, Software Developer, IT Engineer

Module IV Course Content

CS2202::Data Structures

Course Prerequisites: Basic programming Skills (C/C++).

Course Objectives:

1. To introduce the basic concepts of data structures and algorithms.

2. To emphasize concepts about searching and sorting techniques.

3. To construct and implement various data structures and abstract data types including lists, stacks, queues, trees, and graphs.

4. To make understand about writing algorithms and step by step approach in solving problems with the help of fundamental data structures.

5. To associate data structures in developing and implementing efficient algorithms.

Credits:..5..... Teaching Scheme Theory: …3….. Hours/Week

Tut: …1….. Hours/Week

Lab: …2….. Hours/Week

Course Relevance: This is a basic Course for Computer Engineering and allied branches. This course has a high relevance in all domains of computer engineering such as in Industries; research etc. as a basic prerequisite course. Data Structures are a crucial part of computer algorithms as they allow programmers to do data management efficiently. A wise selection of data structures can improve the performance of a computer program or algorithm in a more useful way.

SECTION-I
Arrays, Stacks, Queues and Linked Lists. **Arrays:** Representation and application of Single and Multidimensional arrays, Sparse Matrix. **Sorting Techniques:** Merge Sort, Quick Sort, Heap sort. **Linked Lists:** Dynamic memory allocation, Singly Linked Lists, Doubly linked Lists, Circular linked lists and Generalized linked lists, Applications of Linked list, introduction to

vectors and applications.

Stack: Stack representation and Implementation using arrays and Linked lists. Applications of stack in Recursion, Expression conversions and evaluations.

Queues: Representation and implementation using array and Linked lists, Types of queue. Applications of Queues: Job Scheduling, Josephus problem etc.

SECTION-II

Trees and Graphs and Hashing.

Trees:- Basic terminology, representation using array and linked lists. Tree Traversals: Recursive and Non recursive, Operations on binary tree. Binary Search trees (BST), Threaded Binary Tree.

Graphs: Terminology and representation using Adjacency Matrix and Adjacency Lists, Graph Traversals and Application: BFS and DFS, Connected graph, Bipartite Graph, Detecting Cycle in graph. Minimum Spanning tree: Prims and Kruskal's Algorithm, Shortest Path Algorithms, Union Find.

Searching techniques: Linear Search, Binary search with Analysis.

Hashing: Hashing techniques, Hash table, Hash functions. Collision handling and Collision resolution techniques.

List of Tutorials: (Any Three)

1) Sorting Techniques: Insertion, Merge sort, Bubble, Shell Sort, Radix Sort.

2) Problem solving using stack (Maze problem, Tower of Hanoi).

3) Expression conversion like infix to prefix and postfix and vice versa.

4) Priority Queues and Job Scheduling Algorithm.

5) Generalized Linked Lists.

6) Threaded Binary tree and Stack less Traversals using TBT.

7) AVL and R-B Tree.

8) Applications of Graph in Network problems.

9) Searching Techniques: Ternary Search, Fibonacci Search.

10) Design of Hashing Functions and Collision Resolution techniques.

11) Cuckoo Hashing**.**

List of Practical's: (Any Six)

1) Assignment based on Sorting.

2) Assignment based on Stack Application (Expression conversion etc.)

3) Assignment based on Queue Application (Job scheduling, resources allocation etc.)

4) Assignment based on linked list.

5) Assignment based on BST operations(Create, Insert, Delete and Traversals)

6) Assignment based on various operations on Binary Tree (Mirror image, Height, Leaf node display, Level wise display etc.)

7) Assignment based on AVL tree.

8) Assignment based on DFS and BFS

9) Assignment based on MST using Prim's and Kruskals Algorithm.

10) Assignment based on Finding shortest path in given Graph.

11) Assignment based on Hashing.

List of Projects:

1. Finding Nearest Neighbors.

2. Calendar Application using File handling.

3. Path finder in Maze

4. Word Completion Using Tire.

5. Bloom Filters.

6. Different Management Systems.

7. Scheduling Applications and Simulation.

8. Shortest Path Applications. (Kirchhoff's Circuit, TSP with Scenario.)

9. Efficient Storage and Data Retrieval Systems.

10. Different Gaming Application.

List of Course Seminar Topics:

1. Asymptotic Notations in Data structures.

2. Hash Table, Heaps and Their applications.

3. Analysis of Merge Sort, Quick Sort and Bubble Sort for Best, Average and Worst Case.

4. Solving N-queen and Josephus Problem using Backtracking , Stack and Queue respectively.

5. Priority Queue in Job Scheduling.

6. Application of Stack in Backtracking problems.

7. Priority Heap and min-Max Heap.

8. Data Structures for Languages and Libraries.

9. Multidimensional and Special Data Structures.

10. Algorithm Designing using Divide and Conquer

List of Course Group Discussion Topics:

1. Application based comparison of Sorting Algorithms.

2. Graphs vs Tree Data Structures: Application based comparison? Which is best? Why? How?

3. Advanced trees: which is the best? (AVL,RB,B,B+) when? how? why?

4. Scenario Based Comparison: Kruskals vs Prims Algorithm.

5. Hashing application in today's technology. Is it necessary?

6. Application based comparison: Stack vs Queues.

7. B- Tress VS B+ Trees: Which is to be consider? When ? Why?

8. Need and Role of Different tree Traversals.

9. Graphs vs Tree Data Structures: Application based comparison? Which is best? Why? How?

10. Linked List application in today's technology. Is it necessary?

List of Home Assignments:

Design:

1. Design Single Source multiple destination Shortest Path Algorithm For Driving Application.

2. Expression Tree and Topological Sorting application in Problem solving.

3. Scheduling Algorithms using Queue.

4. Implementation of B and B+ trees for database management.

5. GLL application to Solve problems on Multivariable Polynomial. Consider suitable example.

Case Study:

1. Consider a Suitable Example for Hashing Application. Study its Merits, Demerits and Design.

2. Consider different real life examples where different sorting, Searching techniques have been used. Why used? How? Comparative study.

3. Why there is a need of different tree traversal algorithms? Consider different real life examples where they are used. Why? How?

4. Game Base study for data structures.

5. Compare different graph traversal algorithm by considering different real life examples where they have used.

Blog

1. Comparative Application of Prims vs Kruskals Algorithm in real life scenarios.

2. AVL Tree vs RB Tree with applications

3. Need of different Sorting techniques.

4. How Hashing is useful in recent technologies? Consider any application related to it.

5. Role of Stacks and Queues in problem Solving.

Surveys

1. How application of Graph Search Algorithms (DFS and BFS) is there in recent technologies? Consider some real life technologies.

2. How Advanced Trees Data structure plays important role in Database management?

3. Survey of Data Structures for computer Graphics applications.

4. A survey on different hashing Techniques in programming.

5. Graph algorithms in Network Application.

Suggest an assessment Scheme:

MSE, ESE, GD, Seminar, HA, CVV, Lab Assignment, Course Project.

Text Books:

1. *E. Horwitz , S. Sahani, Anderson-Freed, " Fundamentals of Data Structures in C",*

Second Edition, Universities Press.

2. *Y. Langsam, M.J. Augenstein, A.M.Tenenbaum, "Data structures using C and C++",*

Pearson Education, Second Edition.

3. *Narasimha karumanchi, "Data Structures and Algorithm Made Easy", Fifth Edition, CareerMonk publication.*

Reference Books:

1. *J. Tremblay, P. soresan, "An Introduction to data Structures with applications", TMHPublication, 2nd Edition.*

Moocs Links and additional reading material:

www.nptelvideos.in, www.geeksforgeeks.org

Course Outcomes:

The student will be able to –

1)To interpret and diagnose the properties of data structures with their memory representations and time complexity analysis.(1)

2)To use linear data structures like stacks, queues with their applications.(2)

3)To implement operations like searching, insertion, deletion, traversing mechanism etc. on various data structures with the help of dynamic storage representation.(3)

4)To demonstrate the use of binary tree traversals and to perform various operations on Non-linear data structures.(5)

5) To analyze the Graph data structure and to solve the applications of Graph data structures.(4)

6)To design the appropriate data structure by applying various hashing Techniques.(3)

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO1	PO2	PO4	PO5	PO8, PO11	PO12, PSO1, PSO2
3	2	3	2	2, 1	2, 3, 2

CO attainment levels

CO1 -1, CO2 –2, CO3-3, CO4-5, CO5 -4, CO6-3

Future Courses Mapping:

Following courses can be learned after successful completion of this course: Advanced Data Structures, Design and Analysis of Algorithms, Operating Systems, Compiler Design, Systems Programming, Data Science and similar courses.

Job Mapping: *Data Structures and Algorithm is must necessary part of any programming job. Without Data structures it is not possible to be good in Competitive coding. All Industries always looks for a strong knowledge in Data structures. Without learning this course one can't imagine a job in computer/IT related industries and research.*

FF No. : 654

Syllabus Template

CS2225::Theory of Computation

Course Prerequisites: Discrete Mathematics, Computer Programming.

Course Objectives:

1. To introduce basic concepts such as alphabet, strings, Languages, Decision problems, etc to work with the abstract formal setup
2. To construct deterministic/nondeterministic automata for regular languages to prove non regularity of languages through application of Pumping Lemma and Myhill-Nerode theorem.
3. To understand the role of non-determinism in Automata theory
4. To design Context free grammars, Push down automata for Context Free Languages
5. To comprehend meaning of undecidability in the context of Turing Machine Model

Credits: 5 **Teaching Scheme Theory: 3** Hours/Week

Tut: 1 Hours/Week

Lab: **2** Hours/Week

Course Relevance:

This is a foundational course for Computer Science and Engineering. The central theme of the course is to study what makes certain computational problems very hard and the others easy? Is there some concrete theoretical evidence for the exhibited hardness of the problems? The course explores these questions, first by introducing students to the abstract notion of computation and models of computation. Starting from very simple model of state machines to finally cumulating into the Turing machine model (which is a foundation of modern-day computers), several models in between are studied. For every model, questions such as, which computational problems can be/cannot be solved in the model? how efficiently a problem can be solved in a particular model? various closure properties of model are studied. Throughout the course emphasis is given to proving things with concrete mathematical arguments.

The course is very important for understanding the concept of computation in more abstract set-up. Wherever one wants to formally talk about underlying model, the restrictions imposed by the model, what is the power and limitations of the model, the principles learnt in this course are useful. Due to abstract nature of the course, the principles learnt have wide applicability. The course is an essential prerequisite for several advanced courses such as Computational Complexity, Advanced Algorithms, Foundation of Logic, Quantum Computation, Parallel computation, Circuit Complexity etc. On more applied side: The Automata theoretic models, concept of Context Free Grammar and Pushdown Automata studied in the course are very important for Compiler design. The models discussed during the course have direct applications to several machine learning models, Natural Language processing, Artificial Intelligence, Functional Programming.

Once the student gains expertise in thinking abstractly about underlying models of computation it facilitates in systematic study of any other domain (in computer science or otherwise) which demands logical thinking and abstraction.

This course is also relevant for students who want to pursue research career in theory of computing, computational complexity theory, Natural Language Processing, advanced algorithmic research.

SECTION-1

Topics and Contents

Finite Automata:
Introduction to Automata, Computability and Complexity theory, Automaton as a model of computation, Central Concepts of Automata Theory: Alphabets, Strings, Languages. Decision Problems Vs Languages. Finite Automata, Structural Representations, Deterministic Finite Automata (DFA)-Formal Definition, Simplified notation: State transition graph, transition table, Language of DFA, construction of DFAs for Languages and proving correctness, Product construction, Nondeterministic finite Automata (NFA), NFA with epsilon transition, Language of NFA, Conversion of NFA with epsilon transitions to DFA, Automata with output. Applications and Limitation of Finite Automata.

Regular and Non-Regular Languages:
Regular expression (RE), Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleene's Theorem: Equivalence Regular expressions and DFAs, Closure properties of Regular Languages (union, intersection, complementation, concatenation, Kleene closure), Decision properties of Regular Languages, Applications of Regular expressions. Myhill-Nerode theorem and applications: proving non-regularity, lower bound on number of states of DFA, State Minimization algorithm, Equivalence testing of DFAs. Non-Regular Languages, Revisiting Pigeon-Hole principle, Pumping Lemma for regular Languages.

Context Free Grammars (CFG):
Context Free Grammars: Definition, Examples, Derivation, Languages of CFG, Constructing CFG, correctness proof using induction. Closure properties of CFLs (Union, Concatenation, Kleene closure, reversal). Derivation trees, Ambiguity in CFGs, Removing ambiguity, Inherent ambiguity. Simplification of CFGs, Normal forms for CFGs: CNF and GNF. Decision Properties of CFLs (Emptiness, Finiteness and Membership). Applications of CFG.

SECTION-1I

Topics and Contents

Push Down Automata:
Description and definition, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic, Non-deterministic PDAs, CFG to PDA construction (with proof). Equivalence of PDA and CFG (without proof). Intersection of CFLs and Regular language. Pumping lemma for CFLs, non-Context Free Languages, Context Sensitive Languages, Definition and Examples of Context Sensitive Grammars, Chomsky hierarchy.

Turing Machines:
Basic model, definition, and representation, Instantaneous Description, Language acceptance by TM. Robustness of Turing Machine model and equivalence with various variants: Two-way/One-way infinite tape TM, multi-tape TM, non-deterministic TM, Universal Turing Machines. TM as enumerator. Recursive and Recursively Enumerable languages and their closure properties.

Introduction to Undecidability:
Church-Turing Thesis and intuitive notion of Algorithm. Introduction to countable and uncountable sets (countability of set of natural numbers, integers, rational numbers. Uncountability of set of real numbers, points in plane, set of all binary strings), Encoding for Turing machines and countability of set of all Turing machines. Existence of Turing unrecognizable languages via Cantor's diagonalization. Undecidability of Halting problem. Examples of undecidable problems: Post Correspondence Problem, Hilbert's 10th Problem, Tiling problem (without proof). Example of Turing unrecognizable language. Decision properties of R, RE languages and Rice's theorem.

List of Tutorials: (Any Three)

1) Problem solving based on deterministic and non-deterministic finite automata.
2) Advanced problem solving based on DFA, NFA.
3) Problem solving based on Regular expressions.
4) Problem solving based on Pumping Lemma.
5) Understanding Myhill-Nerode theorem.
6) Advanced problems on Context Free Grammars.
7) Problem solving on Pushdown Automata.
8) Problem solving on Turing Machines.
9) Problem solving on Contability
10) Problem solving on undecidability

List of Practicals: (Any Six)

1.**Problem Solving based on Basic Counting:** Propositional logic, Introduction to proofs: direct, contraposition, contradiction, counterexamples, principle of mathematical induction, strong induction. Proving correctness of programs.

Elementary set theory, relations, functions, basic counting principles, permutations, combinations, generalized permutations and combinations (with/without repetitions, distinguishable/indistinguishable objects), Binomial coefficients and identities. Double counting, combinatorial proof technique, Pigeon-Hole Principle and some applications, Inclusion Exclusion Principle, and applications.

Recurrence relations, modeling using recurrence relations (some examples Fibonacci numbers, Catlan numbers, Derangements, Tower of Hanoi, partitions), generating functions and their application in counting.

2. Problem Solving based on Basic Discrete Probability:

Definition of probability, examples, independence of events, conditional probability, union bound, inclusion exclusion, Bayes' rule, discrete random variables, expectation, variance, linearity of expectation, sum of independent random variables, Markov and Chebyshev inequality, weak law of large numbers, standard distributions (Bernoulli, Binomial, Geometric), coupon collector problem, birthday paradox, probabilistic recurrences. Uniform generation of combinatorial structures. Indicator random variables and their role in algorithm analysis.

3. Problem Solving based on Modular Arithmetic:

Number theory – Integers, division algorithm, divisibility and congruences, gcd and Euclid's Algorithm, extended Euclid's algorithm, application to modular inversion, prime numbers, Euclid's proof for infinitude of primes, unique factorization, Fermat's little theorem, Euler's phi function, Euler's theorem, Chinese remainder theorem, Fast modular exponentiation.

4. Problem Solving based on Graph Theory:

[To be taught in combinatorial perspective] Graphs, different representations, properties of incidence and adjacency matrices, directed/undirected graphs, degree of a vertex, connected components, paths, cycles in graph, Eulerian and Hamiltonian tours, Trees, properties of trees,

Simple combinatorial problem solving based on graphs, bipartite graphs (graph with only odd cycles, 2-colorable graphs), Planar graphs, Euler's theorem for planar graph, Graph colorings, matching in bipartite graphs

List of Course Seminar Topics:
1. NFA Vs DFA
2. Pumping Lemma and Applications
3. Closure properties of Regular languages
4. Decision properties of Regular languages
5. Chomsky hierarchy
6. Application of TOC principles in compiler design
7. Hilbert's 10^{th} problem
8. Context Free and Context sensitive grammars
9. Pumping Lemma for CFL and applications
10. Recursive and Recursively enumerable languages

List of Course Group Discussion Topics:
1. Applications of Automata theory in Compiler design
2. Applications of Automata theory in Natural language processing
3. Undecidability
4. Software testing, why it is very hard?
5. Robustness of Turing machine model
6. Godel's incompleteness theorem
7. Countable and un-countable sets
8. P Vs NP problem
9. Church Turing thesis
10.Models of computation

List of Home Assignments:
Design:
1. Solve 5 challenging problems on NFA, DFA
2. Solve 5 challenging problems on non regular, regular languages
3. Solve 5 challenging problems on Context free grammars, PDAs
4. Solve 5 challenging problems on Turing machines
5. Solve 5 challenging problems on undecidability

Case Study:
1. Randomized algorithms for pattern matching
2. Myhill-Nerode theorem and applications
3. Chomsky-Schützenberger Theorem and Dyck languages
4. Lambda Calculus
5. Hilbert's 10^{th} Problem

Blog
1. Finite Automata
2. Timed Automata and applications
3. Buchi Automata

4. Non-regular languages
5. Contability

Surveys

1. Pattern Matching algorithms
2. Parsers
3. Evolution of models of computations
4. Role of nondeterminism in theory of computation
5. Closure and decision properties of Context free languages

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

MSE: 10% + ESE: 10% + Seminar: 15% Group Discussion: 15% + Home Assignments: 10% + Discrete-Maths evaluation 20% + CVV: 20%

Text Books: ***(As per IEEE format)***

1. Hopcroft J, Motwani R, Ullman, Addison-Wesley, "Introduction to Automata Theory, Languagesand Computation", Second Edition, ISBN 81-7808-347-7.

2. Michael Sipser, Course Technology, "Introduction to Theory of Computation", Third Edition,ISBN-10: 053494728X.
3.. "Discrete Mathematics and its applications" by Kenneth Rosen (William C Brown Publisher)

Reference Books: ***(As per IEEE format)***

1. J. Martin, "Introduction to Languages and the Theory of Computation", Third edition, Tata McGraw-Hill, ISBN 0-07-049939-x, 2003.
2. Daniel I. A. Cohen, "Introduction to Computer Theory", Wiley-Second Edition, ISBN-10 : 04711377

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:
The student will be able to –
1. Infer the applicability of various automata theoretic models for recognizing formal languages.
2. Discriminate the expressive powers of various automata theoretic and formal language theoretic computational models.
3. Illustrate significance of non determinism pertaining to expressive powers of various automata theoretic models.
4. Comprehend general purpose powers and computability issues related to state machines and grammars.
5. Explain the relevance of Church-Turing thesis, and the computational equivalence of Turing machine model with the general purpose computers.
6. Grasp the theoretical limit of computation (independent of software or hardware used) via the concept of undecidability.

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO1	PO2	PO4	PO9	PO12	PSO13
3	3	2	1	2	3

CO attainment levels

CO number	**1**	**2**	**3**	**4**	**5**	**6**
Attainment level	**2**	**3**	**3**	**4**	**5**	**5**

Future Courses Mapping:

Compiler design, Computational Complexity theory, Computability theory, Advanced Algorithms, Natural Language Processing, Artificial Intelligence

Job Mapping:
Wherever one wants to formally talk about underlying model, the restrictions imposed by the model, what is the power and limitations of the model, the principles learnt in this course are useful. Due to abstract nature of the course, the principles learnt have wide applicability, let it be domain of Machine learning, Natural Language processing, Compiler design, Parallel

computation, for each of them having background of Theory of Computation is very useful. If student wants to pursue higher education/ research in Computer Science, this course is must.

FF No. :654

Syllabus Template

CS2226:: Software Engineering

Course Prerequisites: Data Structures

Course Objectives:

1. To summarize capabilities and impact of Software Development Process Models and justify process maturity through application of Software Engineering principles and practices

2. To discriminate competing and feasible system requirements indicating correct real world problem scope and prepare stepwise system conceptual model using stakeholder analysis and requirement validation.

3. To formulate system specifications by analyzing User-level tasks and compose software artifacts using agile principles, practices and Scrum framework

4.To compose system analysis and design specifications indicating logical, physical, deployment, and concurrency viewpoints using object-oriented analysis and design principles and Model Driven Engineering practices using UML-supported modeling tools.

5. To comprehend the nature of design patterns by understanding a small number of examples from different pattern categories and apply these patterns in creating a correct design using design heuristics

6.To propose multi-faceted defendable solutions demonstrating team-skills accommodating design patterns reducing the potential cost and performance impedance in order to realize system artifacts with the help of Model Driven Development practices using, scheduling, estimation and risk management activities.

Credits:.5..... **Teaching Scheme Theory:…3 Hours/Week**

Tut: 1 Hours/Week

Lab:...2.Hours/Week

Course Relevance: Given that software engineering is built upon the foundations of both computer science and engineering, a software engineering curriculum can be approached from either a computer science-first or software engineering-first perspective; there clearly is merit in both approaches. Software engineering spans the entire software lifecycle - it involves creating high-quality, reliable programs in a systematic, controlled, and efficient manner using formal methods for specification, evaluation, analysis and design, implementation, testing and maintenance. any software products are among the most complex of man-made systems, requiring software development techniques and processes that successfully scale to large applications which satisfy timing, size, and security requirements all within acceptable timeframes and budgets. For these reasons, software engineering requires both the analytical and descriptive tools developed in computer science and the rigor that the engineering disciplines bring to the reliability and trustworthiness of the systems that software developers design and implement while working cohesively in a team environment.

SECTION-1

Topics and Contents

Software Engineering Paradigms: Overview of Software Engineering, Software Process Framework, Traditional Process Models, Process Models: Code-and-Fix, Waterfall Model, Rapid Application Development, Incremental Models, Evolutionary Models, Iterative Development, The Unified Process, Cleanroom Methodology, Component-Based Software Engineering, CMMI, Software Engineering Principles and Practices, Requirements Engineering Tasks, Requirement Elicitation Techniques, Software Requirements: Functional, Non-Functional, Domain Engineering activities, Requirements Characteristics and Characterization, Eliminating Requirement Ambiguities, Conflict Identification and Resolution, Requirement Qualities, Requirement Specification, Requirement Traceability, Requirement Prioritization, Relationship of Requirement Engineering to other Framework Activities, System Scope Determination and Feasibility Study, Statement of Work Generation, Requirements Verification and Validation, Requirement Maturity, Technical Reviews, Stakeholder Management

Overview of Agile Methodology: Introducing Agile in Practice, Landscape of Agile and Planned Methods, Agile Challenges in Practice, Composite Agile Method and Strategy (CAMS), Composite Agile and IT: Enablement, Development, and Maintenance, Collaborative-Agile Business Management, Business Analysis and Composite Agile, CAMS Project Management and ICT Governance, Agile Adoption in Organizations. Time-Boxing, Kanban, and Theory of Constraints, Lean IT, Pair Programming, Extreme Programming, DSDM, User Requirements in the context of Agile

The Scrum: Scrum Origins: What Is Scrum? Scrum Origins, Why Scrum? Scrum Framework, Agile Principles, Overview, Variability and Uncertainty, Sprints., Requirements and User Stories, Product Backlog, Estimation and Velocity, Technical Debt, Roles: Product Owner, Scrum Master, Development Team, Scrum Team Structures, Managers, Planning: Scrum Planning Principles, Portfolio Planning, Envisioning (Product Planning), Release Planning (Longer-Term Planning), Sprinting: Sprint Planning, Sprint Execution, Sprint Review, Sprint Retrospective, Scrum and Service Industry

SECTION-1I

Topics and Contents

System Behavior Specification: Static Behavior: Use Cases, Use Case Diagram Components, Use Case Diagram, Actor Generalization, Include and Extend, Template for Use Case Narrative, Building Domain Model, and capturing system behavior in use cases, Use cases and User Stories, Dynamic Behavior: Sequence diagrams, object lifelines and message types, Modeling collections multiobjects, Refining sequence diagrams, Collaboration diagrams, States, events and actions, Nested machines and concurrency, Modifying the object model to facilitate states, Modeling methods with activity diagrams, Activity Diagrams: Decisions and Merges, Synchronization, Iteration, Partitions, Parameters and Pins, Expansion Regions, Swimlanes, concurrency and synchronization, Communication Diagram, Interaction Overview Diagrams, Timing Diagrams

Software Architecture Design and Configuration Management: Analysis Concepts, Analysis Methods, The Design Model, Design Qualities, Characteristics of Design activities, Design Principles, Cohesion and Coupling, Software Architecture Vs Software Design, Software Reuse, Design Heuristics, User Interface Design: Rules, User Interface Analysis and Steps in Interface Design, Design Evaluation, Source Code Management,

Foundations of Software Architecture, Reference Architectures, Architectural Design: Software Architecture, Data Design and Architectural Design, Views, Viewpoints, Perspectives, Conceptual Architecture View, Module Architecture View, Execution Architecture View, Code Architecture View.

Architecture styles: data-flow, object oriented, layered, data-centered, call and return, Repository, Pipe-Filter, Peer-Peer, Publish-Subscribe, Client-Server, Two-Tier, Three-Tier, N-Tier, Heterogeneity in Architecture, Categorizing classes: entity, boundary and control , Modeling associations and collections, Preserving referential integrity, Achieving reusability, Reuse through delegation, Identifying and using service packages, Improving reuse with design Packages and interfaces: Distinguishing between classes/interfaces, Exposing class and package interfaces.

Project Management Principles and Design Patterns:, Design Patterns: Introduction to Design Pattern, Describing Design Patterns, Catalogue of Design Patterns Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton, Structural Patterns: Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy, Behavioral Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor, Antipatterns, Applications of Design Patterns, Project Management Activities, Structures and Frameworks, Teamwork, Leadership, Project Planning, Project

Scheduling, Risk Analysis, Introduction to Function Points, Empirical Estimation, COCOMO II model.

List of Tutorials: (Any Four)

1. Study of Requirement Engineering
2. Study on preparation of System Requirement Specification
3. Scrum Artifacts
4. User Stories and Use Case
5. Product Backlog Development
6. Burn-up and Burn-down chart development and management
7. Software System Analysis and Design: UML
8. Incorporation of Design patterns

List of Practical's: (Any Eight)

1. A real-world problem issue is required to be identified with manageable scope. The problem scenarios are required to be identified for target system to be developed. The scenarios are stated in the form of Statement-of-Work template. The SOW document shall address the vision, goals, and objectives of the project.
2. The initial requirements and feature set for the target system is required to be identified. The requirements are required to be synthesized with stakeholder participation. The project roles are assigned to the project team with clear indicator of responsibilities. The initial requirements summary document with adequate and minimal infrastructure is required to be developed using multiple iterations.
3. The product backlog for the project aimed at maintaining a prioritized queue of project requirements shall be created.
    a. It should be dynamic and should be continuously groomed as the project progresses. Agile projects generally use an iceberg strategy for grooming the product backlog.
    b. The items that are near the top of the iceberg and are closest to going into development should get the most attention.
    c. There should typically be about two to three sprints worth of stories at the top of the backlog that are well-groomed and ready to go into development in order to avoid a situation where the project team is waiting for work to do.
4. Sprint-level planning activity accommodating story points, planning poker shall be performed. The Sprint-plan and Sprint-design indicating detailed activity planner shall be developed.
5. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system. Specify the behavior of the target system and map requirements to Use cases.
    a. The System Context Diagram depicts the overall System behavioral trace and Requirement Capture diagram depicts the hierarchical Use case Organization. The Use Case diagram should encompass
    b. Actors (External Users)
    c. Transactions (Use Cases)
    d. Event responses related to transactions with external agents.
    e. Detection of System boundaries indicating scope of system.
6. To depict the dynamic behavior of the target system using sequence diagram. The Sequence diagram should be based on the Scenarios generated by the inter-object Communication. The model should depict:
    a. Discrete, distinguishable entities (class).
    b. Events (Individual stimulus from one object to another).
    c. Conditional events and relationship representation.
7. To depict the state transition with the life history of objects of a given class model. The

model should depict:

a. Possible ways the object can respond to events from other objects.
b. Determine of start, end, and transition states.

8. To depict the dynamic behavior using detailed Activity diagram. Activity is a parameterized behavior represented as coordinated flow of actions. The flow of execution is modeled as activity nodes connected by activity edges.
    a. A node can be the execution of a subordinate behavior, such as an arithmetic computation, a call to an operation, or manipulation of object contents.
    b. Activities may form invocation hierarchies invoking other activities, ultimately resolving to individual actions.
9. To develop logical static structure of target system with Software Class diagram. To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from System analysis phase. The design model should depict
    a. Relationship between classes: inheritance, Assertion, Aggregation, Instantiation
    b. Identification of objects and their purpose.
    c. Roles / responsibilities entities that determine system behavior.
10. To enhance Software Class diagram to Architecture diagram with appropriate design patterns. The patterns selected shall be justifiable and applied to individual and distinct hierarchies. Suitable Architectural Styles shall be selected and the structural elements shall be well-documented.

To represent physical module that provides occurrence of classes or other logical elements identified during analysis and design of system using Component diagram. The model should depict allocation of classes to modules. To narrate precise Program Design Language constructs separating computation from interface. To represent deployment view of the system through Architecture Diagram.

List of Projects:

1. Automated Parking lot identifier
2. Health Care Software's
3. Financial Domain
4. Appraisal Systems
5. Automate Project Administration System

6. Translator for Agriculture System
7. Development of applications manageable by Agile
8. Development of SMART applications

List of Course Seminar Topics:

1. Agile software development
2. AI and software engineering
3. Apps and app store analysis
4. Automated reasoning techniques
5. Autonomic and (self-)adaptive systems
6. Big data
7. Cloud computing
8. Component-based software engineering
9. Computer-supported cooperative work
10. Configuration management and deployment
11. Crowd sourced software engineering
12. Cyber physical systems
13. Data-driven software engineering
14. Debugging
15. Dependability, safety, and reliability

List of Course Group Discussion Topics:

1. Distributed and collaborative software engineering
2. Domain modelling and meta-modelling
3. Education
4. Embedded software
5. Emerging domains of software
6. Empirical software engineering
7. End-user software engineering
8. Fault localization
9. Formal methods
10. Green and sustainable technologies
11. Human and social aspects of software engineering
12. Human-computer interaction
13. Knowledge acquisition and management
14. Machine learning for software engineering
15. Middleware, frameworks, and API

List of Home Assignments:

Design:

1. Software visualization

2. Specification and modeling languages
3. Tools and environments
4. Traceability
5. Ubiquitous and pervasive software systems
6. Validation and verification

Case Study:

1. Software economics and metrics
2. Software engineering for machine learning
3. Software evolution and maintenance
4. Software modeling and design
5. Software process
6. Software product lines

Blog

1. Mining software engineering repositories
2. Mobile applications
3. Model-driven engineering
4. Parallel, distributed, and concurrent systems
5. Performance
6. Program analysis
7. Program comprehension
8. Program repair
9. Program synthesis
10. Programming languages
11. Recommendation systems
12. Refactoring

Surveys

1. Requirements engineering
2. Reverse engineering
3. Safety-critical systems
4. Scientific computing
5. Search-based software engineering
6. Security, privacy and trust
7. Software architecture
8. Software reuse
9. Software services
10. Software testing

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

MSE+ESE+HA+LAB+CP+CVV+SEMINAR+GD

Text Books: ***(As per IEEE format)***

1. *Ian Sommerville, 'Software Engineering', Addison-Wesley, 9th Edition, 2010, ISBN-13: 978-0137035151.*
2. *Kenneth S. Rubin, Essential SCRUM: A Practical Guide To The Most Popular Agile Process, Addison-Wesley, ISBN-13: 978-0-13-704329-3, 2012*
3. *Tom Pender, "UML Bible", John Wiley & sons, ISBN – 0764526049*

Reference Books: ***(As per IEEE format)***

1. *SorenLauesen, Software requirements: Styles and techniques, Addison Wesley, ISBN 0201745704, 2002*
2. *Dean Leffingwell, Agile Software Requirements, Addison-Wesley, ISBN-13: 978-0-321-63584-6, 2011*
3. *Charles G. Cobb, The Project Manager's Guide To Mastering Agile: Principles and Practices for an Adaptive Approach, Wiley Publications, ISBN: 978-1-118-99104-6 (paperback), ISBN 978-1-118-99177-0 (epdf), 2015*
4. *Grady Booch, James Rambaugh, Ivar Jacobson, "Unified Modeling Language Users Guide", 2nd Edition, Addison- Wesley, ISBN – 0321267974*
5. *Erich Gamma, Richard Helm, Ralph Johnson, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley Professional, ISBN-10: 0201633612 ISBN-13: 978-0201633610*
6. *Paul Clements, Felix Bachmann, Len Bass, David Garlan, Documenting Software Architectures: Views and Beyond Addison-Wesley Professional 2003, ISBN-10:0201703726, ISBN-13: 9780201703726*

Moocs Links and additional reading material: www.nptelvideos.in

www.nptelvideos.in
www.coursera.com

www.udemy.com

Course Outcomes:

1. Summarize capabilities and impact of Software Development Process Models and justify process maturity through application of Software Engineering principles and practices focusing tailored processes that best fit the technical and market demands of a modern software project.
2. Discriminate competing and feasible system requirements indicating correct real world problem scope and prepare stepwise system conceptual model using stakeholder analysis and requirement validation.
3. Formulate system specifications by analyzing User-level tasks and compose software artifacts using agile principles, practices and Scrum framework along with Propose and demonstrate realistic solutions supported by well-formed documentation with application of agile roles, sprint management, and agile architecture focusing project backlogs and velocity monitoring.
4. Compose system analysis and design specifications indicating logical, physical, deployment, and concurrency viewpoints using object-oriented analysis and design principles and Model Driven Engineering practices using UML-supported modeling tools.
5. Comprehend the nature of design patterns by understanding a small number of examples from different pattern categories and apply these patterns in creating a correct design using design heuristics, published guidance, applicability, reasonableness, and relation to other design criteria resulting in well-documented system profiles to the engineering and social community.
6. Propose multi-faceted defendable solutions demonstrating team-skills accommodating design patterns reducing the potential cost and performance impedance in order to realize system artifacts with the help of Model Driven Development practices using, scheduling, estimation and risk management activities.

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO3	PO4	PO8	PO11	PSO3
2	3	3	2	1	3

CO attainment levels

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO3	PO4	PO8	PO11	PSO3
1	5	2	3	3	4

Future Courses Mapping:

Software testing and Quality Assurance, Service-oriented Software

Job Mapping:

Application Architect, Project Designer, SCRUM Role Players

FF No. : 654

CS2227 ::Database Management Systems

Course Prerequisites: Data structures, Discrete Mathematics

Course Objectives:

1. To introduce the fundamentals of different data modeling techniques.
2. To design and development of relational database management systems.
3. To Study the theory behind database systems, the issues that affect their functionality and performance
4. To design of query languages and the use of semantics for query optimization.
5. To understand the latest trends of data management systems.

Credits: 4 **Teaching Scheme Theory: 3** Hours/Week

Tut: NA

Lab: **2** Hours/Week

Course Relevance: The course emphasizes on the fundamentals of database modeling and design, the languages and models provided by the database management systems, and database system implementation techniques. The goal is to provide an in-depth and up-to-date presentation of the most important aspects of database systems and applications, and related technologies.

SECTION-I

Topics and Contents

Introduction: Need of Database Management Systems, Evolution, Database System Concepts and Architecture, Database Design Process
Data Modeling: Entity Relationship (ER) Model, Extended ER Model, Relational Model, Codd's Rules;
Database Design: Need of Normalization, Functional Dependencies, Inference Rules, Functional Dependency Closure, Minimal Cover, Decomposition Properties, Normal Forms: 1NF, 2NF, 3NF and BCNF, Multi-valued Dependency, 4NF, Relational Synthesis Algorithms
Query Languages: Relational Algebra, SQL: DDL, DML, Select Queries, Set, String, Date and Numerical Functions, Aggregate Functions ,Group by and Having Clause, Join Queries, Nested queries, DCL, TCL, PL/SQL: Procedure, Function, Trigger, Mapping of Relational Algebra to SQL

SECTION-II

Topics and Contents

Storage and Querying: Storage and File structures, Indexed Files, Single Level and Multi Level Indexes; Query Processing, Query Optimization

Transaction Management: Basic concept of a Transaction, ACID Properties, State diagram, Concept of Schedule, Serializability – Conflict and View, Concurrency Control Protocols, Recovery techniques

Parallel and Distributed Databases: Architecture, I/O Parallelism, Interquery, Intraquery, Intraoperation and Interoperation Parallelism, Types of **Distributed** Database Systems, Distributed Data Storage, Distributed Query Processing

NOSQL Databases and Big Data Storage Systems: Introduction to NOSQL Databases, Types of NOSQL Databases, BASE properties, CAP theorem, Big Data, HADOOP: HDFS, MapReduce.

Data Warehousing: Architecture and Components of Data Warehouse, Warehouse Schemas, OLAP

List of Practicals: (Any Six)

1) Create a database with appropriate constraints using DDL and populate/modify it with the help of DML.

2) Design and Execute "SELECT" queries using conditional, logical, like/not like, in/not in, between...and, is null/is not null operators in where clause, order by, group by, aggregate functions, having clause, and set operators. Use SQL single row functions for date, time, string etc.

3) Write equijoin, non equijoin, self join and outer join queries. Write queries containing single row / multiple row / correlated sub queries using operators like =, in, any, all, exists etc. Write DML queries containing sub queries. Study a set of query processing strategies.

4) Write PL/SQL blocks to implement all types of cursor.

5) Write useful stored procedures and functions in PL/SQL to perform complex computation.

6) Write and execute all types of database triggers in PL/SQL.

7)Execute DDL statements which demonstrate the use of views. Try to update the base table using its corresponding view. Also consider restrictions on updatable views and perform view creation from multiple tables.

8) Create a database with suitable example using MongoDB and implement Inserting and saving document, Removing document, Updating document

9) Execute at least 10 queries on any suitable MongoDB database that demonstrates following querying techniques:find and findOne, Query criteria, Type-specific queries

10) Implement Map Reduce operation with suitable example using MongoDB.

List of Tutorial (Any 6)

1) Draw ER Diagram for any information system application
2) Draw Extended ER Diagram for any information system application
3) Convert ER Diagram to Relational Model
4) Convert EER Diagram to Relational Model
5) Solving the Queries using Relational Algebra
6) Normalization using Minimal Cover,
7) Normalization using Synthesis Algorithm
8) Creation of Data warehouse for any organisation.
9) OLAP & OLTP
10) Advanced SQL

List of Projects:

Designing and Implementing a Small-scale Relational DBMS
Phase 1: SQL interpreter
Phase 2: Persistent data management
Phase 3: Relational Operations

List of Course Seminar Topics:

1. Object and Object-Relational Databases
2. XML data model, XML documents and associated languages
3. Database Security
4. Modern Storage Architectures
5. Google Cloud- SQL Databases
6. Google Cloud- NOSQL Databases
7. Amazon Databases
8. Oracle NoSQL Database
9. Cassandra DB
10. Data Center Engineering

List of Course Group Discussion Topics:
1. RDBMS Vs NOSQL
2. ER model Vs UML diagrams
3. Normalized vs unnormalized database
4. OLTP Vs OLAP
5. Data Warehouse Vs Data Lake
6. RDBMS and OODBMS
7. Neo4J and GraphBase
8. DynamoDB Vs Voldemort
9. Google File System (GFS) Vs HDFS (Hadoop Distributed File System)
10. Hive SQL Vs Pig Latin

List of Home Assignments:
Design:
1. Suppose you want to build a video site similar to YouTube. Identify disadvantages of keeping data in a file-processing system. Discuss the relevance of each of these points to the storage of actual video data, and to metadata about the video, such as title, the user who uploaded it, tags, and which users viewed it.
2. Illustrate data model that might be used to store information in a social-networking system such as Facebook
3.Describe the circumstances in which you would choose to use embedded SQL rather than SQL alone or only a general-purpose programming language.
4. Give the DTD and XML Schema for Library Management System. Give a small example of data corresponding to this DTD and XML.Write ten queries in Xpath and XQuery
5.If you were designing a Web-based system to make airline reservations and sell airline tickets, which DBMS architecture would you choose? Why? Why would the other architectures not be a good choice?Design a schema and show a sample database for that application. What types of additional information and constraints would you like to represent in the schema? Think of several users of your database, and design a view for each.

Case Study:
1. PostgreSQL
2. Oracle
3. IBM DB2 Universal Database
4. Microsoft SQL Server
5. SQLite database

Blog
1.OLAP tools from Microsoft Corp. and SAP
2.Views in database
3. Dynamic SQL and Embedded SQL
4. Active databases and Triggers
5. SQL injection attack

Surveys

1. Keyword queries used in Web search are quite different from database queries. List key differences between the two, in terms of the way the queries are specified, and in terms of what is the result of a query.
2. List responsibilities of a database-management system. For each responsibility,explain the problems that would arise if the responsibility were not discharged
3. List reasons why database systems support data manipulation using a declarative query language such as SQL, instead of just providing a a library of C or C++ functions to carry out data manipulation
4. Consider a bank that has a collection of sites, each running a database system. Suppose the only way the databases interact is by electronic transfer of money between themselves, using persistent messaging.Would such a system qualify as a distributed database? Why?
5. Data warehousing products coupled with database systems

Suggest an assessment Scheme:

MSE:10 ESE:10 HA:10 CP:10 Lab:10 Seminar:15 GD: 15 CVV:20

Text Books:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan; "Database System Concepts"; 6th Edition, McGraw-Hill Education
2. Ramez Elmasri, Shamkant B. Navathe; "Fundamentals of Database Systems"; 7th Edition, Pearson

Reference Books:

1. Thomas M. Connolly, Carolyn E. Begg," Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition ;Pearson
2. Raghu Ramakrishnan, Johannes Gehrke; "Database Management Systems", 3rd Edition; McGraw Hill Education
3. Kristina Chodorow, MongoDB The definitive guide, O'Reilly Publications, ISBN: 978-93-5110-269-4, 2nd Edition.
4. Dr. P. S. Deshpande, SQL and PL/SQL for Oracle 10g Black Book, DreamTech.
5. Ivan Bayross, SQL, PL/SQL: The Programming Language of Oracle, BPB Publication. 6. Reese G., Yarger R., King T., Williums H, Managing and Using MySQL, Shroff Publishers and Distributors Pvt. Ltd., ISBN: 81 - 7366 - 465 – X, 2nd Edition.
7. Dalton Patrik, SQL Server – Black Book, DreamTech Press.
8. Eric Redmond, Jim Wilson, Seven databases in seven weeks, SPD, ISBN: 978-93-5023-918-6.
9. Jay Kreibich, Using SQLite, SPD, ISBN: 978-93-5110-934-1, 1st edition.

Moocs Links and additional reading material:
https://nptel.ac.in/courses/106/105/106105175/
https://onlinecourses.nptel.ac.in/noc21_cs04/preview
https://www.datacamp.com/courses/introduction-to-sql
Oracle MOOC: PL/SQL Fundamentals - Oracle APEX

Course Outcomes:
1)Design and draw ER and EER diagrams for real life applications.
2)Transform conceptual schema of high level data model into implementation data model
3)Apply the concepts of normalization to develop the quality relational data model
4)Formulate queries in relational algebra, SQL and write PL/SQL blocks.
5)Acquaint with physical database file structures
6)Identify the use of database techniques such as NOSQL

CO PO Map
CO1: PO2 CO2:PO3 CO3:PO5 CO4:PO9 CO5:PO10 CO6:PSO15

CO attainment levels
CO1:1 CO2:4 CO3:2 CO4:4 CO5:3 Co6:5

Future Courses Mapping:
Advanced databases
Big Data Management
Cloud Databases
Database Administrator

Job Mapping:
Database Engineer
SQL developer
PL/SQL developer

FF No. : 654

CS2208::SOFTWARE DEVELOPMENT PROJECT –II

Course Prerequisites: C++ and JAVA

Course Objectives:

1. To develop problem solving ability using programming skills by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students

Credits:.3..... **Teaching Scheme Theory:…. Hours/Week**

Tut: Hours/Week

Lab:..6..Hours/Week

Course Relevance: Software project development comes under the category of project based learning (PBL). PBL is an instructional approach designed to give students the opportunity to develop knowledge and skills through engaging projects set around challenges and problems they may face in the real world by using domain specific language technologies. PBL is "learning by doing."

Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process.

The PBL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy**.**

Teacher's Role in PBL :

- Teacher is not the source of solutions; rather he will act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Selection of Project/Problem:

- The problem-based project oriented model for learning is recommended. The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.
- There are no commonly shared criteria for what constitutes an acceptable project. Projects vary greatly in the depth of the questions explored, the clarity of the learning goals, the content, and structure of the activity.
- Use of technology in meaningful ways to help them investigate, collaborate, analyse, synthesize, and present their learning.
- Activities may include- Solving real life problems, investigation, /study and Writing reports of in depth study, field work.

Student's Role in PBL:

- Students must have the ability to initiate the task/idea .They should not be mere imitators.
- They must learn to think.
- Students working in PBL must be responsible for their own learning.
- . Students must quickly learn how to manage their own learning, Instead of passively

receiving instruction.

- Students in PBL are actively constructing their knowledge and understanding of the situation
- Students in PBL are expected to individually.

Developing Inquiry Skills:

- Students in PBL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution.
- · How effective is …….? How strong is the evidence for ……? How clear is ……?
- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.
- Skills required by students in information literacy include:
- How to prepare the search? How to carry out the research
- Sorting and assessing of information in general

Use of Research Methodology: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

Sample Software Project Statement based on Java and Mobile Application Development

1) Design and deploy an android app for real time criminal detection on the basis of database provided by the police department.

2) Design and deploy an android app for real time health alarm generation like Aarogya Setu.

3) Design and deploy a system for real time home kitchen accidents and appliances control.

...not limited to.....Faculty and students are free to include other area which meets the society requirements at large.

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

Higher levels of the Booms Taxonomy - analyze, apply, evaluate and create.

Text Books: ***(As per IEEE format)***

1. *A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017*
2. *Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.*
3. *Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robart Capraro, Mary Margaret Capraro*

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from societal need point of view

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO1, PO3	PO2, PO8, PO11, PO12, PSO3, PSO4	PO3, PO4, PSO1	PO5, PO8	PO6,7,9,12	PO12, PSO1, PSO2
3, 3	3, 2, 2, 2, 3, 3	2, 2, 1	2, 2	2, 2, 2, 1	2, 3, 3

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

Future Courses Mapping:

Mention other courses that can be taken after completion of this course

Job Mapping:

What are the Job opportunities that one can get after learning this course

Software Engineer, Software Developer, IT Engineer

CS2210::ENGINEERING DESIGN AND INNOVATIONS-IV

Course Prerequisites: Problem Based Learning

Course Objectives:

1. To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students

Credits:.4..... **Teaching Scheme Theory:…. Hours/Week**

Tut: Hours/Week

Lab:..8.Hours/Week

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Engineering Design" are designed as a ladder to extend connectivity of software technologies to solve real word problem using interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Group Structure:

- There should be a team/group of 4-5 students.
- A supervisor/mentor teacher assigned to individual groups.

· It is useful to group students of different abilities and nationalities together.

Selection of Project/Problem:

- Students must focus to initiate the task/idea .The idea inception and consideration shall be from following areas as a real world problem:
- Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
- This is the sample list to start with. Faculty and students are free to include other areas which meet the society requirements at large.
- The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher's Role in PCL :

- Teacher is not the source of solutions rather he will they act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Student's Role in PCL:

- Students must have ability to initiate the task/idea .they should not be mere imitators.
- They must learn to think.
- Students working in PCL must be responsible for their own learning.
- Students must quickly learn how to manage their own learning, Instead of passively receiving instruction.
- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution. How effective is …….? How strong is the evidence for ……? How clear is ……?
- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.

- Skills required by students in information literacy include:
- · How to prepare the search? How to carry out the research
- Sorting and assessing of information in general

Use of Research Methodology: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

EDI Sample Case Studies : -

With the adaptation of industry communication standards, Raspberry Pi and Sensors, following projects can be taken up:

1) Design a deployable product for soil moisture detection

2) Design a deployable product for temperature detection

3) Design a deployable product for pressure detection

3) Design a deployable product smoke detection

4) Design a deployable product for motion detection

5) Design a deployable product for collision detection

6) Design a deployable product for sound detection

...not limited to.....Faculty and students are free to include other areas which meet the society requirements at large.

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

To focus on the higher levels of the Booms Taxonomy analyze, apply, evaluate and create.

Text Books: ***(As per IEEE format)***

1. *A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017*

2. *Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.*

2. *Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robart Capraro, Mary Margaret Capraro*

Reference Books: ***(As per IEEE format)***

1. *De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based learning in engineering. Rotterdam: Sense Publishers. 2007.*

2. *Project management core textbook, second edition, Indian Edition , by Gopalan.*

3. *The Art of Agile Development. By James Shore & Shane Warden.*

Moocs Links and additional reading material: www.nptelvideos.in

https://worldwide.espacenet.com/

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from societal need point of view

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO1	PO2, PO8	PO3	PO4, PO5, PO8	PO6, PO7, PO9, PO12	PO12, PSO1, PSO2, PSO4
3	3, 1	2	2, 2, 2	2, 2, 2, 1	2, 3, 2, 2

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

Job Mapping:

What are the Job opportunities that one can get after learning this course

Software Engineer, Software Developer, IT Engineer

Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Structure & Syllabus of

T.Y. B. Tech. (Computer Engineering)

Effective from Academic Year 2021-22

Prepared by: - Board of Studies in Computer Engineering

Pattern "C21"

T. Y. B. Tech. Computer Engineering AY 2021-22

Module V courses

FF No. : 654

Syllabus Template

CS3205::Design and Analysis of Algorithms

Course Prerequisites: Basic courses on programming, data structures, discrete structures, theory of computing.

Course Objectives:

1. Students will gain understanding of asymptotic notations and will be able to apply suitable mathematical techniques to find asymptotic time and space complexities of algorithms.
2. Students will develop the ability to formulate computational problems in the abstract and mathematically precise manner.
3. Student will gain understanding of different algorithm design paradigms such as divide and conquer, dynamic programming, greedy, backtracking and will apply suitable paradigm for designing algorithms for computational problems
4. Students will develop understanding of notions of NP-hardness and NP-completeness and their relationship with the intractability of decision problems.

1. Students will design randomized, approximation algorithms for some computational problems.

Credits:.5..... **Teaching Scheme Theory:…3 Hours/Week**

Tut: 1 Hours/Week

Lab:...2.Hours/Week

Course Relevance: This is a foundational course for Computer science and Engineering. This course develops algorithmic thinking capability of students. Designing algorithms using suitable paradigm and analysing the algorithms for computational problems has a high relevance in all domains where computer science plays a crucial role (equally in Industry as well as research). This course is also an essential pre-requisite for advanced domain specific algorithmic courses such as Algorithmic Graph Theory, Algorithmic Number Theory, Computational Geometry, Motion planning and Robotics, etc, to give a few examples.

Once the student gains expertise in Algorithm design and in general gains ability of Algorithmic thinking, it facilitates in systematic study of any other domain (in computer science or otherwise) which demands logical thinking.

This course is also relevant for students who want to pursue research career in theory of computing, computational complexity theory, advanced algorithmic research.

SECTION-1

Basic introduction and time and space complexity analysis:

Asymptotic notations (Big Oh, small oh, Big Omega, Theta notations). Best case, average case, and worst-case time and space complexity of algorithms. Overview of searching, sorting algorithms. Adversary lower bounds (for the comparison-based sorting algorithms, for finding second minima). Using Recurrence relations and Mathematical Induction to get asymptotic bounds on time complexity. Master's theorem and applications. Proving correctness of algorithms.

Divide and Conquer: General strategy, Binary search and applications, Analyzing Quick sort, Merge sort, Counting Inversions, Finding a majority element, Order statistics (randomized and deterministic algorithms), Josephus problem using recurrence, Efficient algorithms for Integer arithmetic (Euclid's algorithm, Karatsuba's algorithm for integer multiplication, fast exponentiation).

Dynamic Programming: General strategy, simple dynamic programming based algorithms to compute Fibonacci numbers, binomial coefficients, Matrix Chain multiplication, Optimal binary search tree (OBST) construction, Coin change problem, 0-1 Knapsack, Traveling Salesperson Problem, All pair shortest path algorithm, Longest increasing subsequence problem, Largest independent set for trees.

SECTION-1I

Greedy and Backtracking strategy:

Greedy: General strategy, Analysis and correctness proof of minimum spanning tree and shortest path algorithms, fractional knapsack problem, Huffman coding, conflict free scheduling.

Backtracking: General strategy, n-queen problem, backtracking strategy for some NP-complete problems (e.g. graph coloring, subset sum problem, SUDOKU)

Introduction to complexity classes and NP-completeness:

Complexity classes P, NP, coNP, and their interrelation, Notion of polynomial time many one reductions reduction, Notion of NP-hardness and NP-completeness, Cook-Levin theorem and implication to P versus NP question, NP-hardness of halting problem. NP-Complete problems (some selected examples from - Satisfiability problem, Circuit-SAT, 3-CNF SAT, vertex cover problem, independent set problem, clique problem, Hamiltonian-circuit problem, subset sum problem, Integer Linear Programming.), reducing NP problems to Integer Linear Programming**.**

Introduction to Randomized and Approximation algorithms:

Introduction to randomness in computation, Las-Vegas and Monte-Carlo algorithms, Abundance of witnesses/solutions and application of randomization, solving SAT for formulas with "many" satisfying assignments, randomized quick sort, Las-Vegas and Monte-Carlo algorithms for majority search, Karger's Min-cut algorithm, coupon collector problem, randomized data structures (randomized BST, skip lists)

Introduction to Approximation algorithms for NP-optimization problems, Approximation algorithm for Vertex Cover, metric Traveling-Sales-Person Problem (metric-TSP), Hardness of approximation for TSP.

List of Tutorials: (Any Three)

1) Problem solving based on asymptotic notations, solution of recurrences

2) Problem solving based on Divide and Conquer strategy

3) Advanced problem solving based on Divide and Conquer strategy

4) Problem solving based on Dynamic Programming strategy

5) Advanced problem solving based on Dynamic Programming strategy

6) Problem solving based on Greedy strategy

7) Problem solving based on Backtracking strategy

8) Proving correctness of algorithms: some techniques

9) Adversary lower bound technique

10) Problem solving based on complexity classes, NP-completeness.

11) Randomized Algorithms

12) Approximation Algorithms

List of Practical's: (Any Six)

1) Assignment based on some simple coding problems on numbers, graphs, matrices
2) Assignment based on analysis of quick sort (deterministic and randomized variant)
3) Assignment based on Divide and Conquer strategy (e.g. majority element search, finding kth rank element in an array)
4) Assignment based on Divide and Conquer strategy (e.g. efficient algorithm for Josephus problem using recurrence relations, fast modular exponentiation)
5) Assignment based on Dynamic Programming strategy (e.g. Matrix chain multiplication, Longest increasing subsequence)
6) Assignment based on Dynamic Programming strategy (e,g, All pair shortest path, Traveling Sales Person problem)
7) Assignment based on Greedy strategy (e.g. Huffman encoding)
8) Assignment based on Backtracking (e.g. graph coloring, n-queen problem)
9) Assignment based on Las-Vegas and Monte-Carlo algorithm for majority element search
10) Assignment based on factor-2 approximation algorithm for metric-TSP

List of Projects:

1. Applications of A* algorithm in gaming
2. Pac-Man game
3. File compression techniques
4. Solution of Maze (comparing the backtracking based solution and Dijkstra's algorithm)
5. Different exact and approximation algorithms for Travelling-Sales-Person Problem
6. Creation of Maze using backtracking
7. Knight tour algorithms
8. Network flow optimization and maximum matching
9. AI for different games such as minesweeper, shooting games, Hex, connect-4, sokoban, etc
10. SUDOKU solver
11. Graph theoretic algorithms
12. Computational Geometry Algorithms
13. AKS primality testing
14. Algorithms for factoring large integers
15. Randomized algorithms for primality testing (Miller-Rabin, Solovay-Strassen)
16. Slider puzzle game

List of Course Seminar Topics:

1. Divide and Conquer Vs Dynamic Programming
2. Greedy strategy
3. NP-hardness
4. Backtracking strategy
5. Dynamic Programming Vs Greedy
6. Computational Complexity
7. Philosophical relevance of P Vs NP question
8. Complexity classes
9. Space complexity
10. Compression Techniques
11. Real world applications of Graph theoretic algorithms
12. Approximation algorithms
13. Hardness of approximation
14. Pseudorandom number generators

List of Course Group Discussion Topics:

1. Greedy Algorithms
2. Dynamic Programming strategy
3. Dynamic Programming Vs Greedy
4. NP-completeness
5. P Vs NP question
6. Algorithm design paradigms
7. Different Searching techniques
8. Backtracking strategy
9. Relevance of Cook-Levin theorem
10. Randomness in computation
11. Approximation Algorithms
12. Application of Recursion

List of Home Assignments:

Design:

1. Problem solving based on Divide and Conquer strategy
2. Problem solving based on Dynamic Programming strategy
3. Problem solving based on Greedy strategy
4. Problem solving based on Backtracking strategy
5. Problems on Randomized Algorithms
6. Problems on Approximation Algorithms
7. Problems on NP completeness

Case Study:

1. AKS primality test
2. Quadratic sieve factoring algorithm
3. Huffman Encoding, LZW encoding
4. Network flow optimization algorithms
5. Approximation algorithms for TSP
2. Cook-Levin theorem and its relationship with intractability of computational problems
7. Sorting techniques

Blog

1. Approximation Algorithms
2. Randomized Algorithms
3. Computational Geometry Algorithms

4. Number Theoretic Algorithms
5. Graph Theoretic Algorithms
6. P Vs NP Problem
7. Complexity classes
8. Greedy Algorithms
9. Divide and Conquer Vs Dynamic Programming

Surveys

1. Primality Testing Algorithms
2. Integer Factoring Algorithms
3. NP-complete problems
4. Compression Techniques
5. Shortest Path Algorithms
6. Algorithms for finding Minimum Weight Spanning Tree
7. **3.** SAT solvers

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

Text Books: ***(As per IEEE format)***

1. Cormen, Leiserson, Rivest and Stein "Introduction to Algorithms" ,PHI 3nd edition, 2009. ISBN 81-203-2141-

2. Jon Kleinberg, Eva Tardos "Algorithm Design", Pearson, 1st edition, 2005. ISBN 978-81-317-0310-6

3. Dasgupta, Papadimitriu, Vazirani "Algorithms" McGraw-Hill Education; 1 edition (September 13, 2006), ISBN-10: 9780073523408, ISBN-13: 978-0073523408

Reference Books: ***(As per IEEE format)***

1. Motwani, Raghavan "Randomized Algorithms", Cambridge University Press; 1 edition (August 25, 1995), ISBN-10: 0521474655, ISBN-13: 978-0521474658

2. Vazirani, "Approximation Algorithms", Springer (December 8, 2010), ISBN-10: 3642084699, ISBN-13: 978-3642084690

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:

The student will be able –

1) To formulate computational problems in abstract and mathematically precise manner

2) To design efficient algorithms for computational problems using appropriate algorithmic paradigm

3) To analyze asymptotic complexity of the algorithm for a complex computational problem using suitable mathematical techniques.

4) To establish NP--completeness of some decision problems, grasp the significance of the notion of NP--completeness and its relationship with intractability of the decision problems.

5) To understand significance of randomness, approximability in computation and design randomized algorithms for simple computational problems and design efficient approximation algorithms for standard NP-optimization problems.

6) To incorporate appropriate data structures, algorithmic paradigms to craft innovative scientific solutions for complex computing problems.

CO PO Map:

CO1	CO2	CO3	CO4	CO5	CO6
PO1	PO3,PO4	PO2,PO3	PO6	PO12	PSO1
3	2,3	2,3	2	2	3

CO attainment levels:

CO No.	**1**	**2**	**3**	**4**	**5**	**6**

Attainment level	**1**	**3**	**2**	**3**	**4**	**5**

Future Courses Mapping:

Following courses can be learned after successful completion of this course:

Advanced Algorithms, Computational Complexity, Computational Geometry, Algorithmic Number Theory, Algorithmic Graph Theory

Job Mapping:

Algorithm design lie at heart of any Computer Science/Engineering application. Once the student gains expertise in Algorithm design and in general gains ability of Algorithmic thinking, it facilitates in systematic studying any other domain (in computer science or otherwise) which demands logical thinking. Algorithm design is an essential component of any job based on programming. All Industries in computer Engineering always look for a strong knowledge in Algorithm design and Data structures. If student wants to pursue higher education/ research in Computer Science, this course is must.

Syllabus Template

CS3203::System Software

Course Prerequisites: Computer Organization, Data Structure

Course Objectives:

1. To describe importance of variuos language processors
2. To understand working of different disk scheduling algorithms
3. To exemplify different file management systems
4. To interrelate DOS and BIOS functionality with current OS
5. To design device drivers for Linux.

Credits: 5 **Teaching Scheme**

Theory: 3 Hours/Week **Tut: 1** Hours/Week

Lab: 2 Hours/Week

Course Relevance: This course is a fundamental course in computer engineering curriculum & used as a prerequisite to study advanced courses like distributed systems, parallel computing, etc.

SECTION-I
Introduction to System Software: Introduction, software types, software hierarchy, components of system software, machine structure, interfaces, address space, levels of System software, recent trends in software development. Language processors - fundamentals of language processing, life cycle of a source program, language processing activities. **Introduction to Device Driver**, driver history, Requirements of Device Drivers, Role of Device Drivers, Classes of Devices, Security issues, Design issues, kernel level device drivers, virtual device drivers(VxD), device driver stack buses and physical devices, static device drivers, dynamic device drivers, PnP, device namespace, and named devices. **PCI Bus Drivers**, The Peripheral Component Interconnect (PCI) Interface, Peripheral component interconnect (PCI) addressing, Boot time, Configuration registers and initialization, Linux PCI initialization, Old-style PCI probing, USB Drivers. **I/O and File Management:** I/O Devices - Types, Characteristics devices, OS design issues for I/O management, I/O Buffering. Disk Scheduling: FCFS, SCAN, C File Management: Concepts, File Organization, File Directories, File Sharing. Record Blocking, Secondary Storage Management, Free Space management, Security

SECTION-II

Systems Programming for Linux as Open Source OS: Essential concepts of linux system programming, APIs and ABIs, standards, program segments/sections, the elf format, linking and loading, linux dynamic libraries (shared objects), dynamic linking, API compatibility, dynamically linked libraries.

Advanced system programming concepts: Operating system interfaces, stack smashing. Multitasking and paging, address translation, memory protection, comparison with windows.

Encoding, Decoding: Encoding and decoding schemes for the X-86 processor.

DOS: Internals of DOS, DOS loading, DOS memory map, Internal commands, External commands, command interpreter, POST details, POST sequence, PSP (structure details), '.exe' and '.com' file structures, conversion of .exe to .com file.

BIOS: what and why, BIOS calls: int 10h calls, dos calls: int 21h calls, difference between DOS and BIOS.

List of Tutorials: (Any Three)

1) Disk Scheduling Algorithms
2) File management
3) Device Drivers
4) DOS
5) BIOS

List of Practical: (Any Six)

1) Implementation of Disk Scheduling Algorithms
2) Write a device driver for char device
3) Write a device driver for block device
4) Write a device driver for network device
5) Design a kernel module for linux/ unix
6) Design& Implementation of DLL on Linux shared library.
7) .Write a linux character device driver module that implements the open(), close(), read() and write() system calls for a character device.
8) Design TSR programs for real world applications

List of Projects:

1.Design and implementation of a Multiprogramming Operating System: Stage II
i. Paging
ii. Error Handling
iii. Interrupt Generation and Servicing
iv. Process Data Struc
2.Design and implementation of a Multiprogramming Operating System: Stage III
i. I/O Channels
ii. Multiprogramming
iii. I/O Spooling
3. Design multi programming operating system phase 1 with arithmetic & logical instruction
4.Design multi programming operating system phase 3 without swapping
5. Design multi programming operating system phase 3 with swapping

List of Course Group Discussion Topics:

1. File system in desktop OS & mobile OS.
2. Device Drivers in I/O
3. Input & output devices

List of Home Assignments:

Design:

1. I/O Management
2. File Management
3. Disk Scheduling
4. File Management

Case Study:

1. Process Management in Linux
2.Process Management in Android
3.Memory Management in Linux
4. Memory Management in Android
5.Process Management in Windows

Blog

1. File System of Windows
2.File System of Linux
3.File System of Android
4.File System of iOS
5.Memory Management in Windows

Surveys

1.Mobile OS used in Smart Phones
2. OS used in data centers
3.Distributed OS & applications
4.Device Drivers for various devices

Suggest an assessment Scheme:
MSE(15)+ESE(15)+HA(10)+LAB(10)+CP(10)+CVV(20)+SEMINAR(10)+GD(10)

Text Books: ***(As per IEEE format)***

1. *. D M Dhamdhere; "Systems Programming & Operating Systems"; Tata McGraw Hill Publications, ISBN - 0074635794 2.*
2. *John J Donovan; " Systems Programming " ; Tata Mc-Graw Hill edition , ISBN-13 978-0-07-460482-3*

Reference Books: ***(As per IEEE format)***

3. *Silberschatz A., Galvin P., Gagne G ;"Operating System Principles" 7th Edition John Wiley and Sons.*
4. *YashavantKanetkar; "Unix Shell Programming", 2 nd Edition, BPB Publications.*
5. *Forouzan B. A., Gilberg R. F.; "Unix And Shell Programming", 1 st Edition, Australia Thomson Brooks Cole.*
6. *Achyut S. Godbole ,AtulKahate; "Operating Systems", 3 rd Edition, McGraw Hill.*
7. *Robert Love, " Linux System Programming " ;O'Reilly, ISBN 978-0-596-00958-8*
8. *Mahesh Jadhav; " Easy Linux Device Driver "; HighTechEasy publishing, Second edition.*
1. *Stalling William; "Operating Systems", 6thEdition, Pearson Education.*
2. *Silberschatz A., Galvin P., GagneG.; "Operating System Concepts", 9th Edition, JohnWiley and Sons.*
3. *D M Dhamdhere; "Systems Programming & Operating Systems"; Tata McGraw HillPublications, ISBN – 0074635794*
9. *John J Donovan; " Systems Programming " ; Tata Mc-Graw Hill edition , ISBN-13978-0-07-460482-3*
10. *Ray Duncan; "Advanced MSDOS programming"; Microsoft press*

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:
1) Discriminate among different System software and their functionalities.
2) Understand the language processor.
3) Implement the disk scheduling algorithms
4) Interpret the methods and techniques about instructions Encoding-Decoding
5) Write device drivers for hardware components
6) Design TSR programs for real world applications

CO attainment levels
CO1-2
CO2-2
CO3-3
CO4-2
CO5-1
CO6-3

CO PO Map

CO1- PO2(2)

CO2-PO3(3)

CO3-PO4(3)

CO4- PO10(2)

CO5-PO12(1)

CO6-PSO3(3)

Future Courses Mapping:
Distributed Operating Systems
High Performance Computing
Distributed Computing
Compiler Design

Job Mapping:
System Administrator (Linux)
Quality Engineer
Operating System Analyst
Data Developer

FF No. : 654

Syllabus Template

CS3215::Web Technology

Course Prerequisites: Computer Networks

Course Objectives:

1. To describe most commonly used HTML5 and CSS3 tags and attributes for website development.

2. To associate event handling with HTML5 forms and CSS3 using javascript as a front end technology for website development.

3. To extend HTML5 and CSS3 and javascript front end technologies with PHP and Mysql as a serverside and backend technologies for website development.

4. To simplify website development using REST API and Spring boot as server side technologies.

5. To build single page applications using REACT as a reusable UI component technology as client side technology.

6. To assemble REACT as a front end technology and Node js as a server side technology to develop enterprise applications

Credits:.5..... **Teaching Scheme Theory: 3 Hours/Week**

Tut: 1 Hours/Week

Lab: 2.Hours/Week

Course Relevance: The key technology of the information age is global communication. Web technology is a truly global area of study as it enables global communication with the help of web sites. Web technologies are the backbone of all IT infrastructures and their applications in the world. These technologies and applications often emerge in communication within countries of countries and spread rapidly around the world. The main objective of the course is present the basic web technology concepts that are required for developing web applications. The key technology components are descriptive languages, server side program elements and client side program elements. In addition the course gives specific contents that are beneficial for developing web-based solutions, like relational data-base communication basics and information security principles and approaches. Most of the jobs available in the IT industries are web technology related.

SECTION-1

Topics and Contents:

Front End Tools: Introduction to web technology, internet and www, Web site planning and design issues. HTML5: structure of html document, HTML elements: headings, paragraphs, line break, styles, colors, fonts, links, frames, lists, tables, images and forms, CSS, Bootstrap, XML, JSON. **Client Side Technologies:** JavaScript: Overview of JavaScript, Data types, Control Structures, Arrays, Functions and Scopes, HTML5 forms Validation, Objects in JS, DOM: DOM levels, DOM Objects and their properties and methods, Manipulating DOM. **JQuery**: Introduction to JQuery, Loading JQuery, Selecting elements, changing styles, creating elements, appending elements, removing elements, handling events. **Server Side Technologies:** PHP: Introduction to PHP, Features, sample code, PHP script working, PHP syntax, conditions & Loops, Functions, String manipulation, Arrays & Functions, Form handling, Cookies & Sessions, File Handling, Exception Handling, E-mail validations, MySQL with PHP, AJAX

SECTION-1I

Topics and Contents

Spring Boot: Overview of Spring Boot, Spring Framework, Installing Spring Boot, Build Tool Maven/Gradle/Ant, Core Features, Spring Security, Web Applications, JPA for database connectivity, Working with SQL and NoSQL, Messaging, Testing, Deploying Spring Boot Applications, Monitoring

React: Introduction to React, React component, JSX, Render function, Component API, Component lifecycle, State, Props, Mixins, Component composition, Pass data from parent to child, Pass data from child to parent, Component styling, Forms, Events, Refs, Keys, Router, Flux

Node JS: Introduction to Node JS, Installation of Node JS, Node JS Modules, Node Package Manager (npm), Creating Web server, File System, Express JS, Serving Static Resources, Database connectivity

List of Tutorials: (Any Three)

1) Examples of html and CSS

2) Examples of html form validation

(a)Email Validation

(b)Mob No Validation

3) Examples of Bootstrap and XML

4) Examples of JavaScript and JQuery

5) Examples of PHP

6) Examples of PHP

7) Examples of MySQL

8) Examples of React

9) Examples of Node js

10) Examples of Springboot

List of Practical's: (Any Six)

1) Installation and configuration and testing working of XAMPP server for local host.

2) Develop a basic web page using the HTML tags you learned in class.

(Develop a responsive web page for your CV using multiple column layout.)

3) Create an admission Template form for VIT admission Process? Perform the validation for email and phone no fields

(Develop a responsive web site for your CV having video background for first page and perform the validation using email or mobile number as username and a password of min length 11 consisting at least one uppercase letter one digit and one special character.)

4)Create an one IT company Template with video in Background (The Web Page must be Responsive and the page contains video in Background)

5) Write a JavaScript program to reverse the elements of a given array.

6) Develop a website using toggleable or dynamic tabs or pills with bootstrap and JQuery

7) Assume we have a file named "webtech.txt", write the correct syntax to open and read the file content.

8) Create three MySQL database tables and write php scripts to read, insert & delete data through web interface.

9) Write a program to calculate Electricity bill in PHP

(a) You need to write a PHP program to calculate electricity bill using if-else conditions.

(b) Conditions

For first 50 units – Rs. 3.50/unit

For next 100 units – Rs. 4.00/unit

For next 100 units – Rs. 5.20/unit

For units above 250 – Rs. 6.50/unit

You can use conditional statements

10) Design and implement a website using REST API and Spring Boot.

11) Design and implement a website using REST API, Spring Boot and MySQL/Oracle

12) Design and implement a website using REACT, Spring Boot and MySQL/Oracle

13) Design and implement a website using REACT, Node Js and MySQL/Oracle

List of Course Projects:

1. Develop a Website with NLP as a backend

2. Student Grievance System

3. Workflow Management System for MNC

4. Browser-based Game Website using HTML, CSS, JavaScript, Bootstrap

5. Develop an web application that help to farmers to solve their farming problems

6. GST Billing Software for Small Business

7. Online Crime Reporting System using PHP

8. Develop an Online College Voting System

9. Develop an Online Loan Processing System for Farmers.

List of Course Seminar Topics:

1.Evolution of Web Technology

2. CSS and Bootstrap

3. JavaScript and JQuery

4. JSON and AJA**X**

5. Cookies & Sessions

6.PHP and MySQL

7. Angular JS

8. Java Servlets

9. Java Server Pages (JSP)

10. JSP and Servlets

11. JavaBeans

12. Model-View_Controllor (MVC) Paradigm

13. Web Services

14. Strut

15. SOAP

List of Course Group Discussion Topics:

1. Evolution of web technologies

2.HTML v/s XML

3.JavaScript and PHP

4. 21st Century Web Technologies

5. Sql Vs MangoDB

6. Internet Privacy

7. Ruby

8. Rails

9. Groovy

10. Grails

11. EJB

List of Home Assignments:

Design:

1. Website for restaurant

2. Website for e-book shop

3. Website for on-line music store

4. Website for guest visiting your society

5. Website for web search engine

Case Study:

1. Emerging Web Technologies

2. Databases for Servers

3. A case study of JQuery used in any real-time healthcare web application

4. SOAP

5. Angular versions

Blog

1. Frond End Technologies

2. Client Side Technologies

3. Server Side Technologies

4. Web Services

5. EJB Aplcations

Surveys

1. Survey on Frond End Technologies

2. Survey on Client Side Technologies

3. Survey on Server Side Technologies

4. Component based technologies

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

MSE: 10% + ESE: 10% + Seminar: 15% Group Discussion: 15% + Home Assignments: 10% + Course Project: 10% + Lab evaluation: 10% + CVV: 20%

Text Books: ***(As per IEEE format)***

1.Kumar, A., Web technologies, CRC press, 2019

2.Gupta, R., Internet & Web Technologies, Engineering Handbook, 2019

3.Martin, M.G., Programming for Beginners: 6 Books in 1 – Swift+PHP+Java+Javascript+Html+CSS: Basic Fundamental Guide for Beginners, Independently published, 2018

4.Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, O'Reilly Media; 5th edition, 2018

5.Kohli , S., Web Technologies, PPB Publications, 2015

6. Adam Bretz& Colin J Ihrig, "Full Stack Javascript Development with MEAN", SPD, First Edition 2015, Indian Reprint September 2015

7.GiulioZambon, " Beginning JSP, JSF and Tomcat", Apress Publication, Second Edition, 2013

8. Jeremy McPeak& Paul Wilton," Beginning JavaScript", Wrox Publication, Fifth Edition, 2015

9. Jeffrey C.Jackson, "Web Technologies: A Computer Science Perspective", Second Edition, Pearson Education, 2007, ISBN 978-0131856035.

10. Robert W. Sebesta: Programming the World Wide Web, 4th Edition, Pearson education, 2008

Reference Books: ***(As per IEEE format)***

1. Marty Hall, Larry Brown,"Core Web Programming", Second Edition, Pearson Education, 2001, ISBN 978-0130897930.

2. H.M. Deitel, P.J. Deitel and A.B. Goldberg, "Internet & World Wide Web How To Program", Third Edition, Pearson Education, 2006, ISBN 978-0131752429.

3. Chris Bates: Web Programming Building Internet Applications, 3rd Edition, Wiley India, 2006.

4. Xue Bai et al: The web Warrior Guide to Web Programming, Thomson, 2003.

Moocs Links and additional reading material:

https://www.w3.org/html/

HTML, The Complete Reference http://www.htmlref.com/

http://w3schools.org/

http://php.net/

https://jquery.com/

https://developer.mozilla.org/en-US/docs/AJAX

http://www.tutorialspoint.com/css/

Course Outcomes:

On the completion of course, student will able to

1. Create front end web pages using HTML5 and CSS3 tags and attributes
2. Provide validation mechanism and event handling in a website using javascript as a front end technology
3. Integrate front end with serverside and backend technologies for commercial websites using PHP and Mysql
4. Write Web API/RESTful API application programming interface to communicate with Spring boot as a serverside technology.
5. Build single page applications using REACT as a reusable UI component technology as client side technology and Spring boot and Node Js as server side technologies
6. Design and develop three tier enterprise application using client side, server side and back end technologies

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO3	PO4	PO6	PO11	PSO3
2	2	3	3	1	3

CO attainment levels

Attainment Levels: 1, 2, 3, 4, 4, 5

Future Courses Mapping:

Mention other courses that can be taken after completion of this course

Mobile Application Developmemt

Job Mapping:

What are the Job opportunities that one can get after learning this course

Software Engineer, Web Developer, IT Engineer, UI Developer

FF No. : 654

CS3217::Database Management Systems

Course Prerequisites: Data structures, Discrete Mathematics

Course Objectives:

1. Learn the fundamentals of different data modeling techniques.
2. Design and development of relational database management systems.
3. Study the theory behind database systems, the issues that affect their functionality and performance
4. Design of query languages and the use of semantics for query optimization.
5. Understand the latest trends of data management systems.

Credits: 4 **Teaching Scheme Theory: 3** Hours/Week

Tut: No Tutorial

Lab: **2** Hours/Week

Course Relevance: The course emphasizes on the fundamentals of database modeling and design, the languages and models provided by the database management systems, and database system implementation techniques. The goal is to provide an in-depth and up-to-date presentation of the most important aspects of database systems and applications, and related technologies.

SECTION-I
Topics and Contents **Introduction**: Need of Database Management Systems, Evolution, Database System Concepts and Architecture, Database Design Process **Data Modeling**: Entity Relationship (ER) Model, Extended ER Model, Relational Model, Codd's Rules; **Database Design**: Need of Normalization, Functional Dependencies, Inference Rules, Functional Dependency Closure, Minimal Cover, Decomposition Properties, Normal Forms: 1NF, 2NF, 3NF and BCNF, Multi-valued Dependency, 4NF, Relational Synthesis Algorithms **Query Languages**: Relational Algebra, SQL: DDL, DML, Select Queries, Set, String, Date and Numerical Functions, Aggregate Functions ,Group by and Having Clause, Join Queries, Nested queries, DCL, TCL, PL/SQL: Procedure, Function, Trigger, Mapping of Relational Algebra to SQL

SECTION-II

Topics and Contents

Storage and Querying: Storage and File structures, Indexed Files, Single Level and Multi Level Indexes; Query Processing, Query Optimization

Transaction Management: Basic concept of a Transaction, ACID Properties, State diagram, Concept of Schedule, Serializability – Conflict and View, Concurrency Control Protocols, Recovery techniques

Parallel and Distributed Databases: Architecture, I/O Parallelism, Interquery, Intraquery, Intraoperation and Interoperation Parallelism, Types of **Distributed** Database Systems, Distributed Data Storage, Distributed Query Processing

NOSQL Databases and Big Data Storage Systems: Introduction to NOSQL Databases, Types of NOSQL Databases, BASE properties, CAP theorem, Big Data, HADOOP: HDFS, MapReduce.

Data Warehousing: Architecture and Components of Data Warehouse, Warehouse Schemas, OLAP

List of Practicals: (Any Six)

1) Create a database with appropriate constraints using DDL and populate/modify it with the help of DML.

2) Design and Execute "SELECT" queries using conditional, logical, like/not like, in/not in, between...and, is null/is not null operators in where clause, order by, group by, aggregate functions, having clause, and set operators. Use SQL single row functions for date, time, string etc.

3) Write equijoin, non equijoin, self join and outer join queries. Write queries containing single row / multiple row / correlated sub queries using operators like =, in, any, all, exists etc. Write DML queries containing sub queries. Study a set of query processing strategies.

4) Write PL/SQL blocks to implement all types of cursor.

5) Write useful stored procedures and functions in PL/SQL to perform complex computation.

6) Write and execute all types of database triggers in PL/SQL.

7)Execute DDL statements which demonstrate the use of views. Try to update the base table using its corresponding view. Also consider restrictions on updatable views and perform view creation from multiple tables.

8) Create a database with suitable example using MongoDB and implement Inserting and saving document, Removing document, Updating document

9) Execute at least 10 queries on any suitable MongoDB database that demonstrates following querying techniques:find and findOne, Query criteria, Type-specific queries

10) Implement Map Reduce operation with suitable example using MongoDB.

List of Projects:

Designing and Implementing a Small-scale Relational DBMS
Phase 1: SQL interpreter
Phase 2: Persistent data management
Phase 3: Relational Operations

List of Course Seminar Topics:

1. Object and Object-Relational Databases
2. XML data model, XML documents and associated languages
3. Database Security
4. Modern Storage Architectures
5. Google Cloud- SQL Databases
6. Google Cloud- NOSQL Databases
7. Amazon Databases
8. Oracle NoSQL Database
9. Cassandra DB
10. Data Center Engineering

List of Course Group Discussion Topics:

1. RDBMS Vs NOSQL
2. ER model Vs UML diagrams
3. Normalized vs unnormalized database
4. OLTP Vs OLAP
5. Data Warehouse Vs Data Lake
6. RDBMS and OODBMS
7. Neo4J and GraphBase
8. DynamoDB Vs Voldemort
9. Google File System (GFS) Vs HDFS (Hadoop Distributed File System)
10. Hive SQL Vs Pig Latin

List of Home Assignments:

Design:

1. Suppose you want to build a video site similar to YouTube. Identify disadvantages of keeping data in a file-processing system. Discuss the relevance of each of these points to the storage of actual video data, and to metadata about the video, such as title, the user who uploaded it, tags, and which users viewed it.
2. Illustrate data model that might be used to store information in a social-networking system such as Facebook
3.Describe the circumstances in which you would choose to use embedded SQL rather than SQL alone or only a general-purpose programming language.
4. Give the DTD and XML Schema for Library Management System. Give a small example of data corresponding to this DTD and XML.Write ten queries in Xpath and XQuery
5.If you were designing a Web-based system to make airline reservations and sell airline tickets, which DBMS architecture would you choose? Why? Why would the other architectures not be a good choice?Design a schema and show a sample database for that application. What types of additional information and constraints would you like to represent in the schema? Think of several users of your database, and design a view for each.

Case Study:

1. PostgreSQL
2. Oracle
3. IBM DB2 Universal Database
4. Microsoft SQL Server
5. SQLite database

Blog

1.OLAP tools from Microsoft Corp. and SAP
2.Views in database
3. Dynamic SQL and Embedded SQL
4. Active databases and Triggers
5. SQL injection attack

Surveys

1. Keyword queries used in Web search are quite different from database queries. List key differences between the two, in terms of the way the queries are specified, and in terms of what is the result of a query.
2. List responsibilities of a database-management system. For each responsibility,explain the problems that would arise if the responsibility were not discharged
3. List reasons why database systems support data manipulation using a declarative query language such as SQL, instead of just providing a a library of C or C++ functions to carry out data manipulation
4. Consider a bank that has a collection of sites, each running a database system. Suppose the only way the databases interact is by electronic transfer of money between themselves, using persistent messaging.Would such a system qualify as a distributed database? Why?
5. Data warehousing products coupled with database systems

Suggest an assessment Scheme:
MSE:10 ESE:10 HA:10 CP:10 Lab:10 Seminar:15 GD: 15 CVV:20

Text Books:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan; "Database System Concepts"; 6th Edition, McGraw-Hill Education
2. Ramez Elmasri, Shamkant B. Navathe; "Fundamentals of Database Systems"; 7th Edition, Pearson

Reference Books:

1. Thomas M. Connolly, Carolyn E. Begg," Database Systems: A Practical Approach to Design, Implementation, and Management, 6th Edition ;Pearson
2. Raghu Ramakrishnan, Johannes Gehrke; "Database Management Systems", 3rd Edition; McGraw Hill Education
3. Kristina Chodorow, MongoDB The definitive guide, O'Reilly Publications, ISBN: 978-93-5110-269-4, 2nd Edition.
4. Dr. P. S. Deshpande, SQL and PL/SQL for Oracle 10g Black Book, DreamTech.
5. Ivan Bayross, SQL, PL/SQL: The Programming Language of Oracle, BPB Publication. 6. Reese G., Yarger R., King T., Williums H, Managing and Using MySQL, Shroff Publishers and Distributors Pvt. Ltd., ISBN: 81 - 7366 - 465 – X, 2nd Edition.
7. Dalton Patrik, SQL Server – Black Book, DreamTech Press.
8. Eric Redmond, Jim Wilson, Seven databases in seven weeks, SPD, ISBN: 978-93-5023-918-6.
9. Jay Kreibich, Using SQLite, SPD, ISBN: 978-93-5110-934-1, 1st edition.

Moocs Links and additional reading material:
https://nptel.ac.in/courses/106/105/106105175/
https://onlinecourses.nptel.ac.in/noc21_cs04/preview
https://www.datacamp.com/courses/introduction-to-sql
Oracle MOOC: PL/SQL Fundamentals - Oracle APEX

Course Outcomes:
1)Design and draw ER and EER diagrams for real life applications.
2)Transform conceptual schema of high level data model into implementation data model
3)Apply the concepts of normalization to develop the quality relational data model
4)Formulate queries in relational algebra, SQL and write PL/SQL blocks.
5)Acquaint with physical database file structures
6)Identify the use of database techniques such as NOSQL

CO PO Map
CO1: PO2 CO2:PO3 CO3:PO5 CO4:PO9 CO5:PO10 CO6:PSO15

CO attainment levels
CO1:1 CO2:4 CO3:2 CO4:4 CO5:3 Co6:5

Future Courses Mapping:
Advanced databases
Big Data Management
Cloud Databases
Database Administrator

Job Mapping:
Database Engineer
SQL developer
PL/SQL developer

FF No. : 654

CS3223:ENGINEERING DESIGN AND INNOVATION-V

Course Prerequisites: Problem Based Learning

Course Objectives:

1. To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students.

Credits:.4..... **Teaching Scheme Theory:… Hours/Week**

Tut: Hours/Week

Lab:.8.Hours/Week

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Trends in Engineering Technology" are designed as a ladder to extend connectivity of software technologies to solve real word problem using interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Group Structure:

- There should be a team/group of 4-5 students.
- A supervisor/mentor teacher assigned to individual groups.

· It is useful to group students of different abilities and nationalities together.

Selection of Project/Problem:

- Students must focus to initiate the task/idea .The idea inception and consideration shall be from following areas as a real world problem:
- Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
- This is the sample list to start with. Faculty and students are free to include other areas which meet the society requirements at large.
- The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher's Role in PCL :

- Teacher is not the source of solutions rather he will they act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Student's Role in PCL:

- Students must have ability to initiate the task/idea .they should not be mere imitators.
- They must learn to think.
- Students working in PCL must be responsible for their own learning.
- Students must quickly learn how to manage their own learning, Instead of passively receiving instruction.
- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution. How effective is …….? How strong is the evidence for ……? How clear is ……?
- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning

- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.
- Skills required by students in information literacy include:
- How to prepare the search? How to carry out the research
- Sorting and assessing of information in general

Use of Research Methodology: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

EDI Sample Case Studies : -

1) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis

2) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis for SAM and BAM processing and analysis

3) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis for Gnome NGS processing and analysis

4) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis for population genetics simulation

5) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis for proteomics processing and analysis

...not limited to.....Faculty and students are free to include other area which meets the society requirements at large.

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

Higher levels of the Booms Taxonomy - analyze, apply, evaluate and create.

Text Books: ***(As per IEEE format)***

1. A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017

2. Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.

Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robart Capraro, Mary Margaret Capraro

Reference Books: ***(As per IEEE format)***

1. De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based learning in engineering. Rotterdam: Sense Publishers. 2007.

2. Project management core textbook, second edition, Indian Edition , by Gopalan.

3. The Art of Agile Development. By James Shore & Shane Warden.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from societal need point of view

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO1	PO2, PO4	PO3, PO5	PO4, PO5, PO8	PO6, PO7, PO9, PO11	PO12, PSO2, PSO3, PSO4
3	3, 3	2, 3	2, 2, 2	2, 2, 2, 1	2,3,2,2

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

Job Mapping:

What are the Job opportunities that one can get after learning this course

Software Engineer, Software Developer, IT Engineer

CS3219::Machine Learning

Course Prerequisites: Mathematics and Statistics

Course Objectives:

1. To imbibe the concepts, techniques and building blocks of machine learning.
2. To understand mathematics for implementing machine learning algorithms.
3. To learn the supervised and unsupervised learning techniques.
4. To introduce use of computational learning theory
5. To apply feature reduction on real life problems.
6. To build Machine Learning models and implement in real life scenario for different applications.

Credits:.5..... **Teaching Scheme Theory: 3 Hours/Week**

Tut: 1 Hours/Week

Lab: 2.Hours/Week

Course Relevance: Machine Learning (ML) is currently one of the hottest buzzwords in tech and with good reason. The last few years have seen several techniques that have previously been in the realm of science fiction slowly transformed into reality. The importance of ML has been increasing as a growing number of companies are using these technologies to improve their products and services, evaluate their business models, and enhance their decision-making process.

SECTION-1

Introduction of Machine Learning: What is Machine Learning, Types of Learning: Supervised, Unsupervised, Reinforcement. Learning System, Well posed learning problem, Designing a learning system, Issues in machine learning. Hypothesis Space, Hypothesis functions, Hypothesis Evaluation, Bias, Variance, Underfitting, Overfitting, Inductive bias, Evaluation, Training, Testing, Cross-validation. : Error Analysis, Error Metrics, Precision and recall.

Supervised Learning:

Regression: Linear regression with one variable, Cost function, Target Function, Gradient Descent, Gradient Descent For Linear Regression, Linear Regression with Multiple Variables, Multiple Features, Gradient Descent for Multiple Variables, Features and Polynomial Regression, Normal Equation, Normal Equation Noninvertibility.

Decision Tree Learning: Representation, Basic decision tree learning algorithm, Issues in decision tree learning

Logistic Regression: Problem of Overfitting, Logistic Regression, Classification, Hypothesis Representation, Decision Boundary, Cost Function, Simplified Cost Function and Gradient Descent, Advanced Optimization, Multiclass Classification, Problem of Overfitting, Regularized Linear Regression, Regularized Logistic Regression.

Instance based learning: Instance based learning, KNN: k-nearest neighbor learning. Feature Selection, Feature Extraction, Feature reduction, Dimensionality Reduction, Principal Component Analysis (PCA), Collaborative filtering based recommendation,

Bayesian Learning : Probability, Bayesian Learning: Bayes theorem, Maximum likelihood hypothesis, minimum description length principle, Gibbs algorithm, Bayesian belief networks.

Support Vector machine (SVM): Optimization Objective, Large Margin Intuition, Mathematics Behind Large Margin Classification, Kernel functions, Linear SVM, Nonlinear SVM. Genetic algorithm

SECTION-2

Topics and Contents

Neural network for supervised and unsupervised learning: Perceptron, Single layer and multilayer perceptrons, Multilayer network, Cost Function, Backpropagation algorithm, Backpropagation Intuition, Unrolling Parameters, Gradient Checking, Artificial Neural Network: Activation Functions, Learning Rules, Mc-Culloch-pitts Neuron model, Heb net, Introduction to deep neural network

Unsupervised Learning: Clustering: K-means algorithm, Nearest Neighbor Learning, Locally weighted regression, adaptive hierarchical clustering, Gaussian mixture model, Collaborative Filtering, Applications and future scope. Gaussian Process Time series forecasting: encoder-decoder approach as in Deep AR, Correlated time series: High-dimensional multivariate forecasting with low-rank Gaussian Copula Processes, Uncertainty Estimation and out of distribution detection

Computational Learning theory: PAC learning model, Sample complexity for finite hypothesis spaces, Sample complexity for infinite hypothesis spaces, Mistake bound model of learning, VC Dimension, Ensemble learning, Adaboost algorithm, Analytical learning, combined Inductive and analytical learning, Reinforcement Learning. Genetic Algorithm,

Learning set of rules.

Scalable Machine Learning: Scalable learning techniques, such as streaming machine learning techniques; parallel infrastructures such as map-reduce; practical techniques for reducing the memory requirements for learning methods, such as feature hashing and Bloom filters; and techniques for analysis of programs in terms of memory, disk usage, and (for parallel methods) communication complexity.

List of Tutorials: (Any Three)

1 Examples on Distance Metrics

2 Examples on Linear regression with one variable

3 Examples on Linear regression with multiple variables

4 Examples on Decision Tree Algorithm

5 Examples on Logistic Regression

6 Examples on K-NN Algorithm

7 Examples on PCA Algorithm

8 Examples on SVM Algorithm

9 Examples on backpropagation Algorithm

10 Examples on K-means Algorithm

11 Examples on VC Algorithm

12 Examples on Computational Learning Algorithm

13 Examples on Genetic Algorithm

List of Practical's: (Any Six)

1. Assignment on distance metrics in machine learning
Minkowski Distance
Manhattan Distance

Euclidean Distance
Chebychev Distance
Cosine Distance
Mahalanobis Distance

2. Assignment on Linear Regression:

The following table shows the results of a recently conducted study on the correlation of the number of hours spent driving with the risk of developing acute backache. Find the equation of the best fit line for this data

Number of hours spent driving (x)	Risk score on a scale of 0-100 (y)
10	95
9	80
2	10
15	50
10	45
16	98
11	38
16	93

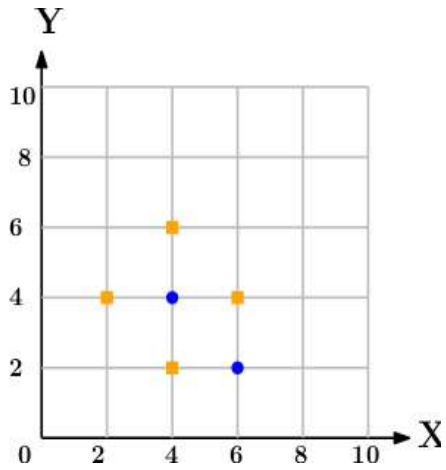
3.Assignment on Decision Tree Classifier:

A dataset collected in a cosmetics shop showing details of customers and whether or not they responded to a special offer to buy a new lip-stick is shown in table below. Use this dataset to build a decision tree, with Buys as the target variable, to help in buying lip-sticks in the future. Find the root node of decision tree. According to the decision tree you have made from previous training data set, what is the decision for the test data: [Age < 21, Income = Low, Gender = Female, Marital Status = Married]?

ID	Age	Income	Gender	Marital Status	Buys
1	< 21	High	Male	Single	No
2	< 21	High	Male	Married	No
3	21-35	High	Male	Single	Yes
4	>35	Medium	Male	Single	Yes
5	>35	Low	Female	Single	Yes
6	>35	Low	Female	Married	No
7	21-35	Low	Female	Married	Yes
8	< 21	Medium	Male	Single	No
9	<21	Low	Female	Married	Yes
10	> 35	Medium	Female	Single	Yes
11	< 21	Medium	Female	Married	Yes
12	21-35	Medium	Male	Married	Yes
13	21-35	High	Female	Single	Yes
14	> 35	Medium	Male	Married	No

4. Assignment on k-NN Classification:

In the following diagram let blue circles indicate positive examples and orange squares indicate negative examples. We want to use k-NN algorithm for classifying the points. If k=3, find the class of the point (6,6). Extend the same example for Distance-Weighted k-NN and Locally weighted Averaging



5. Assignment on Naive Bayes Classifier

Using following dataset answer the following question

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Using Naïve Bayes classifier what will be the class when a new species holds following

feature values:

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

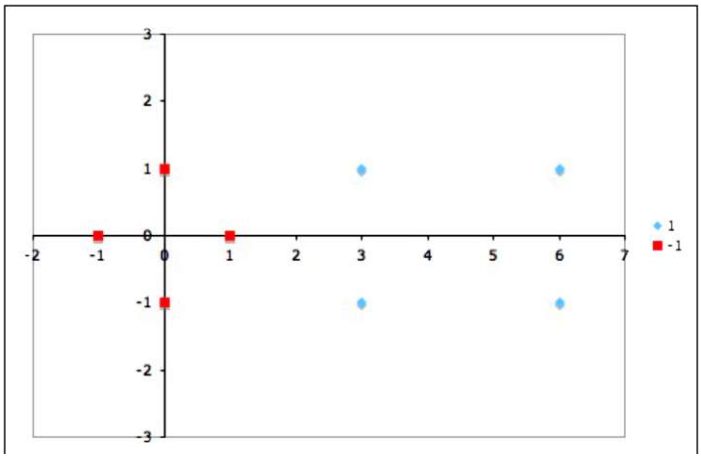
6. Assignment on Linear SVM

Calculate the optimal hyper plane and margin for small dataset shown in following diagram. Positively labeled data points are

$$\left\{\begin{pmatrix}3\\1\end{pmatrix},\begin{pmatrix}3\\-1\end{pmatrix},\begin{pmatrix}6\\1\end{pmatrix},\begin{pmatrix}6\\-1\end{pmatrix}\right\}$$

and negatively labeled data points are

$$\left\{\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}0\\-1\end{pmatrix},\begin{pmatrix}-1\\0\end{pmatrix}\right\}$$



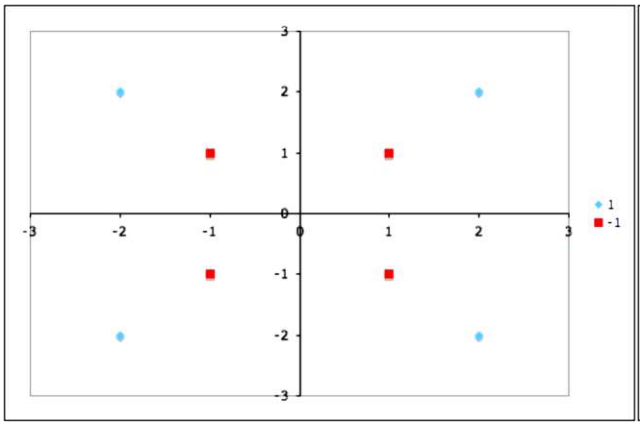
7. Assignment on Non-Linear SVM

Calculate the optimal hyper plane and margin for small dataset shown in following diagram. Positively labeled data points are

$$\left\{\begin{pmatrix}2\\2\end{pmatrix},\begin{pmatrix}2\\-2\end{pmatrix},\begin{pmatrix}-2\\-2\end{pmatrix},\begin{pmatrix}-2\\2\end{pmatrix}\right\}$$

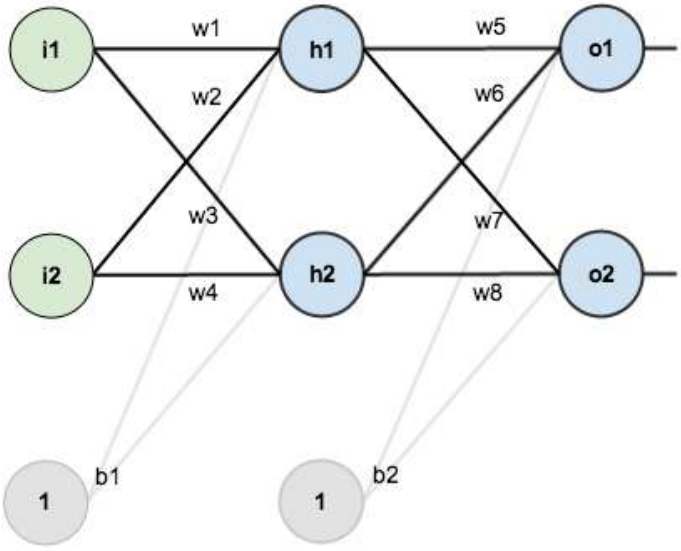
and negatively labeled data points are

$$\left\{\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}1\\-1\end{pmatrix},\begin{pmatrix}-1\\-1\end{pmatrix},\begin{pmatrix}-1\\1\end{pmatrix}\right\}$$

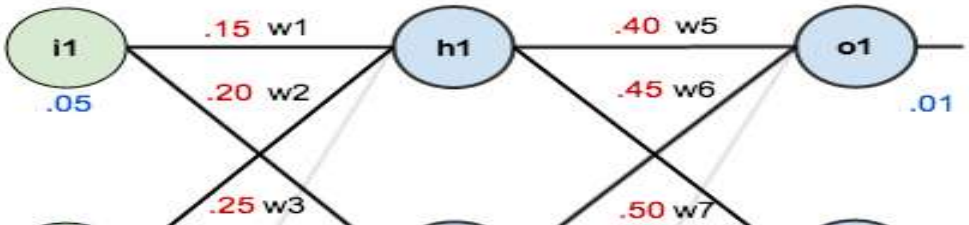


8. Assignment on Backpropagation algorithm:

Use a neural network with two inputs, two hidden neurons, two output neurons. Additionally, the hidden and output neurons will include a bias



In order to have some numbers to work with, here are the initial weights, the biases, and training inputs/outputs:



The goal of backpropagation is to optimize the weights so that the neural network can learn how to correctly map arbitrary inputs to outputs.

9. Assignment on K-Means Clustering:

We have given a collection of 8 points. P1=[0.1,0.6] P2=[0.15,0.71] P3=[0.08,0.9] P4=[0.16, 0.85] P5=[0.2,0.3] P6=[0.25,0.5] P7=[0.24,0.1] P8=[0.3,0.2]. Perform the k-mean clustering with initial centroids as m1=P1 =Cluster#1=C1 and m2=P8=cluster#2=C2. Answer the following

1] Which cluster does P6 belongs to?
2] What is the population of cluster around m2?
3] What is updated value of m1 and m2?

10. Assignment on Ensemble Learning

Given a set of 4 points (x1,-), (x2,-), (x3,+) and (x4,+), Adaboost algorithm is used to train a weak classifier on this data. In the first iteration, the weak classifier wrongly classifies x3 and correctly classifies the other three points. In the second iteration it wrongly classifies x1 and correctly classifies the other points. Assuming uniform initial weight distribution (D1) over the data points, what is the weight distribution for the 3rd iteration (D3).

List of Course Projects: USE the UCI ML or KDD Repository or any other dataset

1 Stock Market Price Prediction
2 Housing Prices Prediction

3 Sign Text Identification
4 Iris Flowers Classification
5 Fake News Detection
6 Product Delivery Drones
7 Smart City Water / Light Management System
8 Human Tracking System
9 Automatic Interview Conduction System
10 Student Information Chatbot Project.
11 Product Review Analysis For Genuine Rating.
12 Customer Targeted E-Commerce
13 MNIST Digit Classification
14 Bit coin Price Predictor
15 Credit Card Fraud Detection
16 Customer Segmentation

List of Course Seminar Topics:

1. Streaming Machine Learning Techniques
2. Error Analysis In Machine Learning
3. Parallel Infrastructures Such As Map-Reduce For ML
4. Ensemble Learning
5. Backpropagation Algorithm.
6. Practical Techniques For Reducing The Memory Requirements: Feature Hashing
7. Support Vector Machine
8. Practical Techniques For Reducing The Memory Requirements: Bloom Filters
9. Genetic Algorithm
10. Regression Analysis
11. K Means Algorithm
12. Decision Tree Learning Algorithm
13. Bayesian Learning, Bayes Theorem And Naïve Bayes Theorem.
14. Hidden Markov Model,
15. Principal Component Analysis (PCA)
16. Recommender Systems

List of Course Group Discussion Topics:

1 Machine Learning And Artificial Intelligence
2 Machine Learning And Data Science
3 Machine Learning Applications
4 Machine Learning Future
5 Machine Learning After 10 Years / 2030
6 Supervised Learning Techniques And Unsupervised Learning Techniques

7 Reinforcement Learning
8 Recommender Systems
9 Will Automation And ML Reduce Or Increase Jobs.
10 Cashless Economy Using ML
11 ML In Covid-19 Situations

List of Home Assignments:

Design:

1. Heart Disease Prediction Using Machine Learning Algorithms
2. Detection Based Project For Social Cause
3. Classification Based Project For Social Cause
4. Clustering Based Project For Social Cause
5. Optimization Based Project For Social Cause
6. Recommender Systems Based Project For Social Cause
7. Identification Based Project For Social Cause
8. Machine Learning-Based Student's Native Place Identification For Real-Time

Case Study:

1. How Auto industry Is Preparing For The 4th Industrial Revolution using ML
2. How Indian Retail Giant Is Using ML to Prepare For The 4th Industrial Revolution
3. Rolls-Royce And Google Partner To Create Smarter, Autonomous Ships Based On ML
4. The Amazing Ways Tesla Is Using Machine Learning And Big Data
5. The Incredible Ways John Deere Is Using Machine Learning To Transform Farming

Blog

1. Machine Learning for Sentiment Analysis
2. Machine Learning for Character Recognition
3. Machine Learning for Heart Disease Detection
4. Machine Learning for Chatbot Development
5. Machine Learning for Agriculture
6. Machine Learning for Medical Field

Surveys

1. Adaption of Machine Learning ML AI in 2020
2. Machine Learning in Industry

3. Machine Learning in Digital Marketing
4. Machine Learning in Military
5. Machine Learning after Covid-19

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

HA, PPT,GD,MSE,ESE,LAB,CVV

Text Books: (As per IEEE format)

1. T. Mitchell, " Machine Learning", McGraw-Hill, 1997.
2. Anup Kumar Srivastava, Soft Computing, Alpha Science International limited. 2009.

Reference Books: (As per IEEE format)

1. Ethem Alpaydin, "Introduction to Machine Learning", MIT press, 2004.
2. Jacek M. Zurada, "Introduction to Artificial neural System", JAICO publishing house,2002,.

Moocs Links and additional reading material:

www.nptelvideos.in

www.coursera.com

Course Outcomes:

1. Formulate a given problem for the given application using machine learning basic building blocks(1)
2. Carryout mathematical analysis of given problem based on learning approach(2)
3. Select appropriate of learning algorithms for sustainable solutions of applications (3)
4. Build a machine learning model for the application (4)
5. Evaluate machine learning modes algorithms(5)
6. Provide scalable machine learning solution(5)

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO1	PO2	PO3	PO5	PO11	PSO3
2	2	3	3	1	3

CO attainment levels

Attainment Levels: 1, 2, 3,4,5,5

Future Courses Mapping:

Mention other courses that can be taken after completion of this course

Soft Computing, Deep Learning

Job Mapping:

What are the Job opportunities that one can get after learning this course

ML Scientist, ML Designer, ML Architectural Design, ML Developer, ML Data Analyst, ML AI Developer

FF No. : 654

Syllabus Template

CS3220::CYBER SECURITY

Course Prerequisites: Computer Networks

Course Objectives:

1. To discover the programming bugs that will be malicious code, they also must be able to explain various attacks and resolve the bugs to mitigate the treats.
2. To interpret various cryptographic techniques to secure the systems developed.
3. To build and explain various authentication and authorization methods with the access control
4. To articulate the use of various standard security protocols for the layered architecture.
5. To state the urgent need for cyber security in critical computer systems, networks, and World Wide Web, and explain various threat scenarios
6. To identify cyber threats and appropriate solutions to critical infrastructures

Credits: 5 **Teaching Scheme Theory: 3** Hours/Week

Tut: 1 Hours/Week

Lab: 2 Hours/Week

Course Relevance: Cyber Security study teach you how to protect computer operating systems, networks, and data from cyber attacks. It covers how to monitor systems and mitigate threats when they happen. This is an oversimplification of IT security degrees curricula. The overall goal is to help you develop the computing skills needed to prevent attacks and protect people's data and privacy.

SECTION-1

Topics and Contents

Introduction to Security : Key security properties - Confidentiality, Integrity, Availability. Risk Management, Understanding Governance- Policies, Framework, Laws, Regulations, Guidelines and Compliance, Risk based Management

Cryptography:

Private key cryptography: Mathematical background for cryptography: modulo arithmetic, GCD (Euclids algorithm), algebraic structures (Groups, Rings, Fields, Polynomial Field). Role of random numbers & nonce in security, Importance of prime numbers

Data Encryption Standard: Block cipher, Stream cipher, Feistel structure, round function, block cipher modes of operation, S-DES, Attacks on DES, S-AES, AES.

Public key cryptography: RSA: RSA algorithm, Key generation in RSA, attacks on RSA. Diffie-Hellman key exchange: Algorithm, Key exchange protocol, Attack. Elliptic Curve Cryptography (ECC): Elliptic Curve over real numbers, Elliptic Curve over Zp, Elliptic Curve arithmetic. Diffie-Hellman key exchange using ECC. Chinese remainder theorem**.**

Data Integrity: Certificates, Introduction to Hashing, Properties of Hash function, HASH + SALT, hashing algorithms (MD5, SHA1, SHA2), Digital Signature, PKI.
Wireless Security: Modes and Protocols for wireless communication, WEP, WPA 2, Hacking Wireless signals, Satellite communication, GSM communication
Fundamentals of Pen-testing: Web, IoT, Mobile.
Authentication and Authorization: Network Access Control: Network Access Control, Extensible Authentication Protocol, IEEE 802.1X Port-Based Network Access Control, SHA-512, Kerberos, X.509 authentication service, Multifactor authentication
IP Security, Session Management, Web Security, Database Security, File Security, Mobile Security
Transport-Level Security: Web Security Considerations, Secure Sockets Layer, Transport Layer Security, HTTPS standard, Secure Shell (SSH) application, IPsec and VPN Tunnelling and Proxies
Email security: PGP and SMIME

SECTION-1I

Topics and Contents

Introduction to Security Attacks and Hacking: Introduction to Security: Vulnerabilities, Threats, Threat Modeling, Risk, attack and attack types, countermeasures - Avoiding attacks, Security services. Protocol Vulnerabilities: DoS and DDoS, session hijacking, ARP spoofing, Pharming attack, Dictionary Attacks. Software vulnerabilities: Phishing, buffer overflow, Cross-site scripting attack, Virus and Worm Features, Trojan horse, Social engineering attacks, ransomware, SYN-Flooding, SQL- injection, DNS poisoning, Sniffing, Introduction to Ethical Hacking, Anonymity, Information Gathering, Scanning Networks, Vulnerability Analysis, Operating System Hacking, Hacking Wireless Networks,
Cloud Hacking, IoT Hacking, OWASP Top 10 and MS STRIDE frameworks for threat assessment, Introduction to SDL (Secure Development Lifecycle) – Merging Security into SDLC, Secure App development (fundamental measures, API integration, etc topics), SIEM (Security Incident and Event Management) & SOAR (Security Operations and Response)
Secure Engineering – Security Assessment and Testing, Vulnerability Assessment and Penetration Testing (VAPT), SANS 25, PSIRT, CVE, MITRE ATT&CK Framework

Introduction to Digital Forensics, Data Recovery and OS Forensics, Email Crimes and Violations,
Cyber Forensics: Cyber Attack, Cyber Reconnaissance, Crimes in Cyber Space-Global Trends

& classification, e-commerce security, Computer forensics, facebook forensic, mobile forensic, cyber forensic, digital forensic,

IoT Security: Definitions of OT, IoT, IIoT, & ICS), Convergence of OT/IoT and IT domains. Functional difference between security measures for IT and OT/IoT, Introduction to most widely used protocols in IoT environment - MQTT and CoAP, KNX, BACnet, BLE, LoRa, ZigBee, SNMP , Embedded/ Hardware device security, Attack vectors and attack surfaces for IoT devices, Layered and Dynamic security measures for IoT Industrial Ecosystem, Introduction to IoT security Standards and its importance, Side Channel Attacks

Fundamentals of Cloud Security: AWS, GCP, and Azure security fundamentals, Secure integration of cloud with IT and IoT systems

Cyber Physical System Attacks and Social Engineering

List of Tutorials:

1. Study of Snort
2. Nessus: a Security Vulnerability scanning tool
3. OWASP Zed Attack Proxy (ZAP).
4. Source Code Analysis Tools
5. Metasploit/Ollydbg
6. Testing SSL
7. Testing for Brute Force Password
8. Testing for SQL Injection

List of Practicals: (Any Six)

1) Acquisition of System Information/ RAM/Volume Shadow Copy/Detecting Encryption in information.
2) Forensic of Disc Image/ Registry/ Meta data/ RAM
3) Simplified DES implementation
4) Simplified AES implementation
5) Encryption and Decryption by RSA algorithm
6) Implementation of ECC over Diffie Hellman Key Exchange Protocol
7) Implementation of authentication algorithms
8) Digital forensic of images
9) Forensics of Video alteration
10) Vulnerabilities finding in Mobile/ computer/ digital devices

List of Projects:

1. Design a System to develop a analyzer which will differentiate between different vulnerability and packets entered using it. This system will detect the intrusions coming through the vulnerabilities.

2. Securing Video Conferencing App for online meetings
3. Steganography for Image/Video/Files
4. Secure Image display on online social media.
5. Secure transfer of government subsidies to farmers/BPL people/ students etc
6. Authentication of users for various applications for integrity, availability, confidentiality.
7. Implementing a system for detecting the modification of videos/images on social media
8. Secure App for online exams detecting Keystroke and camera movements.
9. A system to detect the difference between the voice edited in the audio/video
10. A System to check the vulnerabilities in the websites.

List of Course Seminar Topics:

1. Blcokchain architecture and its implementation
2. Cloud Security
3. Mobile Security
4. IoT and Security Issues/ Security Models for IoT
5. Darkweb
6. Docker Security
7. Access control methods for online social media and various organizations
8. Security of Android Vs IOS
9. Machine learning and SCADA Security
10. Security Applications for Smart Cities

List of Course Group Discussion Topics:

1. Security Issues in Android and IOS devices
2. Industry 4.0 and security
3. Blockchain and E-voting system
4. Security of Aadhar Card and other digital cards
5. Automated Home Appliances and Security
6. Programming Bugs and Malicious code in information security
7. Indian Cyber laws and Deficiencies
8. Social Media and Cyber Security
9. Child abuse on online social media and security
10. Need of cyber crime and security in school education.

List of Home Assignments:

Design:

1. Design a secure system using cryptography techniques for security of multimedia files.
2. Design a secure system using steganography for hiding data files in image/video
3. Design a system for educational institutes using authentication and authorization techniques, also give details about the access control policies that must be implemented for the design of system by various places.
4. Design a secure system using SSL/TLS/IPSec for the various organizations
5. Design a system for the analysis of cyber crime using various cyber forensic techniques and compare each technique with respect to integrity, confidentiality, availability

Case Study:

1. How to improve the security of social media? Write a detail case study
2. Find out the vulnerability issues in educational institutes websites/online systems and give solutions to these problem. Perform a detailed case study of the various issues.
3. Write a detail case study about the banking security flows and solutions to these flows.
4. Give a detail case study of the antivirus system giving the flows and solutions to it.
5. Perform the detail case study of various operating systems used for mobile devices and give a secure solution to one for widely used OS.

Blog

1. Dark Web
2. Crypto currency and Economy
3. Cyber crime and solutions
4. Authentication and Access control for social media
5. Cyber forensic and Cyber laws

Surveys

1. Survey on various blockchain related issues/ cryptocurrency/ application systems developed using blockchain
2. Survey on various authentication and access control methods for different applications
3. Steganography and Biometric Systems for authentication
4. Survey of various attacks and its effect on Indian economy and its analysis
5. Problems over Integer Lattices: A Study

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

MSE: 10% + ESE: 10% + Seminar: 15% Group Discussion: 15% + Home Assignments: 10% + Course Project: 10% + Lab evaluation: 10% + CVV: 20%

Text Books: ***(As per IEEE format)***

1. *William Stallings; "Cryptography and Network Security-Principles and Practices" 6th Edition , Pearson Education, 2014, ISBN13:9780133354690.*
2. *Bernard Menezes, "Network Security and Cryptography", 1st Edition, Cengage Learning, 2010, ISBN 81-315-1349-1.*
3. *Raef Meeuwisse, "Cybersecurity for Beginners", 2nd Edition, Cyber Simplicity, 2017, ISBN-9781911452157*

Reference Books: ***(As per IEEE format)***

1. *M. Speciner, R. Perlman, C. Kaufman, "Network Security: Private Communications in a Public World", Prentice Hall, 2002*
2. *Michael Gregg, "The Network Security Test Lab: A Step-By-Step Guide", Dreamtech Press, 2015, ISBN-10:8126558148, ISBN-13: 978-8126558148.*
3. *Matt Bishop, "Computer Security: Art and Science", 1st Edition, Pearson Education, 2002, ISBN 0201440997.*
4. *Charlie Kaufman, Radia Perlman and Mike Spencer, "Network security, private communication in a public world", 2nd Edition, Prentice Hall, 2002, ISBN 9780130460196.*
5. *V.K. Pachghare, "Cryptography and Information Security", 2nd Edition, PHI, 2015, ISBN-978-81-203-5082-3.*

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:

1) Analyze cryptographic techniques using a mathematical approach by examining nature of attack.
2)Identify and establish different attacks on the system
3) Justify various methods of authentication and access control for application of technologies to various sections of industry and society.
4) Design a secure system for protection from the various attacks for 7 layer model by determining the need of security from various departments of an organization
5) Estimate future needs of security for a system by researching current environment on a continuous basis for the benefit of society.
6) Analyze various types of cyber crime by detecting the crime

CO PO Map

Programme Outcomes												Program Specific Outcome			
PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3	PSO 4
	2	3		3	2			1						3	

CO attainment levels

Future Courses Mapping:

Blockchain Technologies
Ethical Hacking and Network Defense
Cloud and Data Security

Job Mapping:
Security Engineer/Network Security Engineer
Information Security Analyst
Cyber Security Analyst
Cyber Security Associate
Manager-Information Security Secvices
Security Consultant
Penetration Testing Engineer

Module VI Courses

FF No. : 654

Syllabus Template

CS3216::Theory of Computation

Course Prerequisites: Discrete mathematics and Computer Programming

Course Objectives:

1. To introduce basic concepts such as alphabet, strings, Languages, Decision problems, etc to work with the abstract formal setup
2. To construct deterministic/nondeterministic automata for regular languages to prove non regularity of languages through application of Pumping Lemma and Myhill-Nerode theorem.
3. To understand the role of non-determinism in Automata theory
4. To design Context free grammars, Push down automata for Context Free Languages
5. To comprehend meaning of undecidability in the context of Turing Machine Model

Credits: 5 **Teaching Scheme Theory: 3** Hours/Week

Tut: 1 Hours/Week

Lab: **2** Hours/Week

Course Relevance:

This is a foundational course for Computer Science and Engineering. The central theme of the course is to study what makes certain computational problems very hard and the others easy? Is there some concrete theoretical evidence for the exhibited hardness of the problems? The course explores these questions, first by introducing students to the abstract notion of computation and models of computation. Starting from very simple model of state machines to finally cumulating into the Turing machine model (which is a foundation of modern-day computers), several models in between are studied. For every model, questions such as, which computational problems can be/cannot be solved in the model? how efficiently a problem can be solved in a particular model? various closure properties of model are studied. Throughout the course emphasis is given to proving things with concrete mathematical arguments.

The course is very important for understanding the concept of computation in more abstract set-up. Wherever one wants to formally talk about underlying model, the restrictions imposed by the model, what is the power and limitations of the model, the principles learnt in this course are useful. Due to abstract nature of the course, the principles learnt have wide applicability. The course is an essential prerequisite for several advanced courses such as Computational Complexity, Advanced Algorithms, Foundation of Logic, Quantum Computation, Parallel computation, Circuit Complexity etc. On more applied side: The Automata theoretic models, concept of Context Free Grammar and Pushdown Automata studied in the course are very important for Compiler design. The models discussed during the course have direct applications to several machine learning models, Natural Language processing, Artificial Intelligence, Functional Programming.

Once the student gains expertise in thinking abstractly about underlying models of computation it facilitates in systematic study of any other domain (in computer science or otherwise) which demands logical thinking and abstraction.

This course is also relevant for students who want to pursue research career in theory of computing, computational complexity theory, Natural Language Processing, advanced algorithmic research.

SECTION-1

Topics and Contents

Finite Automata:
Introduction to Automata, Computability and Complexity theory, Automaton as a model of computation, Central Concepts of Automata Theory: Alphabets, Strings, Languages. Decision Problems Vs Languages. Finite Automata, Structural Representations, Deterministic Finite Automata (DFA)-Formal Definition, Simplified notation: State transition graph, transition table, Language of DFA, construction of DFAs for Languages and proving correctness, Product construction, Nondeterministic finite Automata (NFA), NFA with epsilon transition, Language of NFA, Conversion of NFA with epsilon transitions to DFA, Automata with output. Applications and Limitation of Finite Automata.

Regular and Non-Regular Languages:
Regular expression (RE), Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleene's Theorem: Equivalence Regular expressions and DFAs, Closure properties of Regular Languages (union, intersection, complementation, concatenation, Kleene closure), Decision properties of Regular Languages, Applications of Regular expressions. Myhill-Nerode theorem and applications: proving non-regularity, lower bound on number of states of DFA, State Minimization algorithm, Equivalence testing of DFAs. Non-Regular Languages, Revisiting Pigeon-Hole principle, Pumping Lemma for regular Languages.

Context Free Grammars (CFG):
Context Free Grammars: Definition, Examples, Derivation, Languages of CFG, Constructing CFG, correctness proof using induction. Closure properties of CFLs (Union, Concatenation, Kleene closure, reversal). Derivation trees, Ambiguity in CFGs, Removing ambiguity, Inherent ambiguity. Simplification of CFGs, Normal forms for CFGs: CNF and GNF. Decision Properties of CFLs (Emptiness, Finiteness and Membership). Applications of CFG.

SECTION-1I

Topics and Contents

Push Down Automata:
Description and definition, Language of PDA, Acceptance by Final state, Acceptance by empty stack, Deterministic, Non-deterministic PDAs, CFG to PDA construction (with proof). Equivalence of PDA and CFG (without proof). Intersection of CFLs and Regular language. Pumping lemma for CFLs, non-Context Free Languages, Context Sensitive Languages, Definition and Examples of Context Sensitive Grammars, Chomsky hierarchy.

Turing Machines:
Basic model, definition, and representation, Instantaneous Description, Language acceptance by TM. Robustness of Turing Machine model and equivalence with various variants: Two-way/One-way infinite tape TM, multi-tape TM, non-deterministic TM, Universal Turing Machines. TM as enumerator. Recursive and Recursively Enumerable languages and their closure properties.

Introduction to Undecidability:
Church-Turing Thesis and intuitive notion of Algorithm. Introduction to countable and uncountable sets (countability of set of natural numbers, integers, rational numbers. Uncountability of set of real numbers, points in plane, set of all binary strings), Encoding for Turing machines and countability of set of all Turing machines. Existence of Turing unrecognizable languages via Cantor's diagonalization. Undecidability of Halting problem. Examples of undecidable problems: Post Correspondence Problem, Hilbert's 10th Problem, Tiling problem (without proof). Example of Turing unrecognizable language. Decision properties of R, RE languages and Rice's theorem.

List of Tutorials: (Any Three)

1) Problem solving based on deterministic and non-deterministic finite automata.
2) Advanced problem solving based on DFA, NFA.
3) Problem solving based on Regular expressions.
4) Problem solving based on Pumping Lemma.
5) Understanding Myhill-Nerode theorem.
6) Advanced problems on Context Free Grammars.
7) Problem solving on Pushdown Automata.
8) Problem solving on Turing Machines.
9) Problem solving on Contability
10) Problem solving on undecidability

List of Practicals: (Any Six)

1.**Problem Solving based on Basic Counting:** Propositional logic, Introduction to proofs: direct, contraposition, contradiction, counterexamples, principle of mathematical induction, strong induction. Proving correctness of programs.

Elementary set theory, relations, functions, basic counting principles, permutations, combinations, generalized permutations and combinations (with/without repetitions, distinguishable/indistinguishable objects), Binomial coefficients and identities. Double counting, combinatorial proof technique, Pigeon-Hole Principle and some applications, Inclusion Exclusion Principle, and applications.

Recurrence relations, modeling using recurrence relations (some examples Fibonacci numbers, Catlan numbers, Derangements, Tower of Hanoi, partitions), generating functions and their application in counting.

2. Problem Solving based on Basic Discrete Probability:

Definition of probability, examples, independence of events, conditional probability, union bound, inclusion exclusion, Bayes' rule, discrete random variables, expectation, variance, linearity of expectation, sum of independent random variables, Markov and Chebyshev inequality, weak law of large numbers, standard distributions (Bernoulli, Binomial, Geometric), coupon collector problem, birthday paradox, probabilistic recurrences. Uniform generation of combinatorial structures. Indicator random variables and their role in algorithm analysis.

3. Problem Solving based on Modular Arithmetic:

Number theory – Integers, division algorithm, divisibility and congruences, gcd and Euclid's Algorithm, extended Euclid's algorithm, application to modular inversion, prime numbers, Euclid's proof for infinitude of primes, unique factorization, Fermat's little theorem, Euler's phi function, Euler's theorem, Chinese remainder theorem, Fast modular exponentiation.

4. Problem Solving based on Graph Theory:

[To be taught in combinatorial perspective] Graphs, different representations, properties of incidence and adjacency matrices, directed/undirected graphs, degree of a vertex, connected components, paths, cycles in graph, Eulerian and Hamiltonian tours, Trees, properties of trees,

Simple combinatorial problem solving based on graphs, bipartite graphs (graph with only odd cycles, 2-colorable graphs), Planar graphs, Euler's theorem for planar graph, Graph colorings, matching in bipartite graphs

List of Course Seminar Topics:

1. NFA Vs DFA
2. Pumping Lemma and Applications
3. Closure properties of Regular languages
4. Decision properties of Regular languages
5. Chomsky hierarchy
6. Application of TOC principles in compiler design
7. Hilbert's 10^{th} problem
8. Context Free and Context sensitive grammars
9. Pumping Lemma for CFL and applications
10. Recursive and Recursively enumerable languages

List of Course Group Discussion Topics:

1. Applications of Automata theory in Compiler design
2. Applications of Automata theory in Natural language processing
3. Undecidability
4. Software testing, why it is very hard?
5. Robustness of Turing machine model
6. Godel's incompleteness theorem
7. Countable and un-countable sets
8. P Vs NP problem
9. Church Turing thesis
10. Models of computation

List of Home Assignments:

Design:

1. Solve 5 challenging problems on NFA, DFA
2. Solve 5 challenging problems on non regular, regular languages
3. Solve 5 challenging problems on Context free grammars, PDAs
4. Solve 5 challenging problems on Turing machines
5. Solve 5 challenging problems on undecidability

Case Study:

1. Randomized algorithms for pattern matching
2. Myhill-Nerode theorem and applications
3. Chomsky-Schützenberger Theorem and Dyck languages
4. Lambda Calculus
5. Hilbert's 10^{th} Problem

Blog

1. Finite Automata
2. Timed Automata and applications
3. Buchi Automata

4. Non-regular languages
5. Contability

Surveys

1. Pattern Matching algorithms
2. Parsers
3. Evolution of models of computations
4. Role of nondeterminism in theory of computation
5. Closure and decision properties of Context free languages

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

MSE: 10% + ESE: 10% + Seminar: 15% Group Discussion: 15% + Home Assignments: 10% + Discrete-Maths evaluation 20% + CVV: 20%

Text Books: ***(As per IEEE format)***

1. Hopcroft J, Motwani R, Ullman, Addison-Wesley, "Introduction to Automata Theory, Languagesand Computation", Second Edition, ISBN 81-7808-347-7.

2. Michael Sipser, Course Technology, "Introduction to Theory of Computation", Third Edition,ISBN-10: 053494728X.
3.. "Discrete Mathematics and its applications" by Kenneth Rosen (William C Brown Publisher)

Reference Books: ***(As per IEEE format)***

1. J. Martin, "Introduction to Languages and the Theory of Computation", Third edition, Tata McGraw-Hill, ISBN 0-07-049939-x, 2003.
2. Daniel I. A. Cohen, "Introduction to Computer Theory", Wiley-Second Edition, ISBN-10 : 04711377

Moocs Links and additional reading material: www.nptelvideos.in

Course Outcomes:
The student will be able to –
1. Infer the applicability of various automata theoretic models for recognizing formal languages.
2. Discriminate the expressive powers of various automata theoretic and formal language theoretic computational models.
3. Illustrate significance of non determinism pertaining to expressive powers of various automata theoretic models.
4. Comprehend general purpose powers and computability issues related to state machines and grammars.
5. Explain the relevance of Church-Turing thesis, and the computational equivalence of Turing machine model with the general purpose computers.
6. Grasp the theoretical limit of computation (independent of software or hardware used) via the concept of undecidability.

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO1	PO2	PO4	PO9	PO12	PSO13
3	3	2	1	2	3

CO attainment levels

CO number	**1**	**2**	**3**	**4**	**5**	**6**
Attainment level	**2**	**3**	**3**	**4**	**5**	**5**

Future Courses Mapping:

Compiler design, Computational Complexity theory, Computability theory, Advanced Algorithms, Natural Language Processing, Artificial Intelligence

Job Mapping:
Wherever one wants to formally talk about underlying model, the restrictions imposed by the model, what is the power and limitations of the model, the principles learnt in this course are useful. Due to abstract nature of the course, the principles learnt have wide applicability, let it be domain of Machine learning, Natural Language processing, Compiler design, Parallel

computation, for each of them having background of Theory of Computation is very useful. If student wants to pursue higher education/ research in Computer Science, this course is must.

FF No. :654

Syllabus Template

CS3218:: Software Modeling and Design

Course Prerequisites: Data Structures

Course Objectives:

1. To summarize capabilities and impact of Software Development Process Models and justify process maturity through application of Software Engineering principles and practices

2. To discriminate competing and feasible system requirements indicating correct real world problem scope and prepare stepwise system conceptual model using stakeholder analysis and requirement validation.

3. To formulate system specifications by analyzing User-level tasks and compose software artifacts using agile principles, practices and Scrum framework

4.To compose system analysis and design specifications indicating logical, physical, deployment, and concurrency viewpoints using object-oriented analysis and design principles and Model Driven Engineering practices using UML-supported modeling tools.

5. To comprehend the nature of design patterns by understanding a small number of examples from different pattern categories and apply these patterns in creating a correct design using design heuristics

6.To propose multi-faceted defendable solutions demonstrating team-skills accommodating design patterns reducing the potential cost and performance impedance in order to realize system artifacts with the help of Model Driven Development practices using, scheduling, estimation and risk management activities.

Credits:.5..... **Teaching Scheme Theory:…3 Hours/Week**

Tut: 1 Hours/Week

Lab:...2.Hours/Week

Course Relevance: Given that software engineering is built upon the foundations of both computer science and engineering, a software engineering curriculum can be approached from either a computer science-first or software engineering-first perspective; there clearly is merit in both approaches. Software engineering spans the entire software lifecycle - it involves creating high-quality, reliable programs in a systematic, controlled, and efficient manner using formal methods for specification, evaluation, analysis and design, implementation, testing and maintenance. any software products are among the most complex of man-made systems, requiring software development techniques and processes that successfully scale to large applications which satisfy timing, size, and security requirements all within acceptable timeframes and budgets. For these reasons, software engineering requires both the analytical and descriptive tools developed in computer science and the rigor that the engineering disciplines bring to the reliability and trustworthiness of the systems that software developers design and implement while working cohesively in a team environment.

SECTION-1

Topics and Contents

Software Engineering Paradigms: Overview of Software Engineering, Software Process Framework, Traditional Process Models, Process Models: Code-and-Fix, Waterfall Model, Rapid Application Development, Incremental Models, Evolutionary Models, Iterative Development, The Unified Process, Cleanroom Methodology, Component-Based Software Engineering, CMMI, Software Engineering Principles and Practices, Requirements Engineering Tasks, Requirement Elicitation Techniques, Software Requirements: Functional, Non-Functional, Domain Engineering activities, Requirements Characteristics and Characterization, Eliminating Requirement Ambiguities, Conflict Identification and Resolution, Requirement Qualities, Requirement Specification, Requirement Traceability, Requirement Prioritization, Relationship of Requirement Engineering to other Framework Activities, System Scope Determination and Feasibility Study, Statement of Work Generation, Requirements Verification and Validation, Requirement Maturity, Technical Reviews, Stakeholder Management

Overview of Agile Methodology: Introducing Agile in Practice, Landscape of Agile and Planned Methods, Agile Challenges in Practice, Composite Agile Method and Strategy (CAMS), Composite Agile and IT: Enablement, Development, and Maintenance, Collaborative-Agile Business Management, Business Analysis and Composite Agile, CAMS Project Management and ICT Governance, Agile Adoption in Organizations. Time-Boxing, Kanban, and Theory of Constraints, Lean IT, Pair Programming, Extreme Programming, DSDM, User Requirements in the context of Agile

The Scrum: Scrum Origins: What Is Scrum? Scrum Origins, Why Scrum? Scrum Framework, Agile Principles, Overview, Variability and Uncertainty, Sprints., Requirements and User Stories, Product Backlog, Estimation and Velocity, Technical Debt, Roles: Product Owner, Scrum Master, Development Team, Scrum Team Structures, Managers, Planning: Scrum Planning Principles, Portfolio Planning, Envisioning (Product Planning), Release Planning (Longer-Term Planning), Sprinting: Sprint Planning, Sprint Execution, Sprint Review, Sprint Retrospective, Scrum and Service Industry

SECTION-1I

Topics and Contents

System Behavior Specification: Static Behavior: Use Cases, Use Case Diagram Components, Use Case Diagram, Actor Generalization, Include and Extend, Template for Use Case Narrative, Building Domain Model, and capturing system behavior in use cases, Use cases and User Stories, Dynamic Behavior: Sequence diagrams, object lifelines and message types, Modeling collections multiobjects, Refining sequence diagrams, Collaboration diagrams, States, events and actions, Nested machines and concurrency, Modifying the object model to facilitate states, Modeling methods with activity diagrams, Activity Diagrams: Decisions and Merges, Synchronization, Iteration, Partitions, Parameters and Pins, Expansion Regions, Swimlanes, concurrency and synchronization, Communication Diagram, Interaction Overview Diagrams, Timing Diagrams

Software Architecture Design and Configuration Management: Analysis Concepts, Analysis Methods, The Design Model, Design Qualities, Characteristics of Design activities, Design Principles, Cohesion and Coupling, Software Architecture Vs Software Design, Software Reuse, Design Heuristics, User Interface Design: Rules, User Interface Analysis and Steps in Interface Design, Design Evaluation, Source Code Management,

Foundations of Software Architecture, Reference Architectures, Architectural Design: Software Architecture, Data Design and Architectural Design, Views, Viewpoints, Perspectives, Conceptual Architecture View, Module Architecture View, Execution Architecture View, Code Architecture View.

Architecture styles: data-flow, object oriented, layered, data-centered, call and return, Repository, Pipe-Filter, Peer-Peer, Publish-Subscribe, Client-Server, Two-Tier, Three-Tier, N-Tier, Heterogeneity in Architecture, Categorizing classes: entity, boundary and control , Modeling associations and collections, Preserving referential integrity, Achieving reusability, Reuse through delegation, Identifying and using service packages, Improving reuse with design Packages and interfaces: Distinguishing between classes/interfaces, Exposing class and package interfaces.

Project Management Principles and Design Patterns:, Design Patterns: Introduction to Design Pattern, Describing Design Patterns, Catalogue of Design Patterns Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype, Singleton, Structural Patterns: Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy, Behavioral Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor, Antipatterns, Applications of Design Patterns, Project Management Activities, Structures and Frameworks, Teamwork, Leadership, Project Planning, Project

Scheduling, Risk Analysis, Introduction to Function Points, Empirical Estimation, COCOMO II model.

List of Tutorials: (Any Four)

9. Study of Requirement Engineering
10. Study on preparation of System Requirement Specification
11. Scrum Artifacts
12. User Stories and Use Case
13. Product Backlog Development
14. Burn-up and Burn-down chart development and management
15. Software System Analysis and Design: UML
16. Incorporation of Design patterns

List of Practical's: (Any Eight)

11. A real-world problem issue is required to be identified with manageable scope. The problem scenarios are required to be identified for target system to be developed. The scenarios are stated in the form of Statement-of-Work template. The SOW document shall address the vision, goals, and objectives of the project.
12. The initial requirements and feature set for the target system is required to be identified. The requirements are required to be synthesized with stakeholder participation. The project roles are assigned to the project team with clear indicator of responsibilities. The initial requirements summary document with adequate and minimal infrastructure is required to be developed using multiple iterations.
13. The product backlog for the project aimed at maintaining a prioritized queue of project requirements shall be created.
    a. It should be dynamic and should be continuously groomed as the project progresses. Agile projects generally use an iceberg strategy for grooming the product backlog.
    b. The items that are near the top of the iceberg and are closest to going into development should get the most attention.
    c. There should typically be about two to three sprints worth of stories at the top of the backlog that are well-groomed and ready to go into development in order to avoid a situation where the project team is waiting for work to do.
14. Sprint-level planning activity accommodating story points, planning poker shall be performed. The Sprint-plan and Sprint-design indicating detailed activity planner shall be developed.
15. To decompose and organize the problem domain area into broad subject areas and identify the boundaries of problem/system. Specify the behavior of the target system and map requirements to Use cases.
    a. The System Context Diagram depicts the overall System behavioral trace and Requirement Capture diagram depicts the hierarchical Use case Organization. The Use Case diagram should encompass
    b. Actors (External Users)
    c. Transactions (Use Cases)
    d. Event responses related to transactions with external agents.
    e. Detection of System boundaries indicating scope of system.
16. To depict the dynamic behavior of the target system using sequence diagram. The Sequence diagram should be based on the Scenarios generated by the inter-object Communication. The model should depict:
    a. Discrete, distinguishable entities (class).
    b. Events (Individual stimulus from one object to another).
    c. Conditional events and relationship representation.
17. To depict the state transition with the life history of objects of a given class model. The

model should depict:

a. Possible ways the object can respond to events from other objects.
b. Determine of start, end, and transition states.

18. To depict the dynamic behavior using detailed Activity diagram. Activity is a parameterized behavior represented as coordinated flow of actions. The flow of execution is modeled as activity nodes connected by activity edges.
    a. A node can be the execution of a subordinate behavior, such as an arithmetic computation, a call to an operation, or manipulation of object contents.
    b. Activities may form invocation hierarchies invoking other activities, ultimately resolving to individual actions.
19. To develop logical static structure of target system with Software Class diagram. To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from System analysis phase. The design model should depict
    a. Relationship between classes: inheritance, Assertion, Aggregation, Instantiation
    b. Identification of objects and their purpose.
    c. Roles / responsibilities entities that determine system behavior.
20. To enhance Software Class diagram to Architecture diagram with appropriate design patterns. The patterns selected shall be justifiable and applied to individual and distinct hierarchies. Suitable Architectural Styles shall be selected and the structural elements shall be well-documented.

To represent physical module that provides occurrence of classes or other logical elements identified during analysis and design of system using Component diagram. The model should depict allocation of classes to modules. To narrate precise Program Design Language constructs separating computation from interface. To represent deployment view of the system through Architecture Diagram.

List of Projects:

9. Automated Parking lot identifier
10. Health Care Software's
11. Financial Domain
12. Appraisal Systems
13. Automate Project Administration System

14. Translator for Agriculture System
15. Development of applications manageable by Agile
16. Development of SMART applications

List of Course Seminar Topics:

16. Agile software development
17. AI and software engineering
18. Apps and app store analysis
19. Automated reasoning techniques
20. Autonomic and (self-)adaptive systems
21. Big data
22. Cloud computing
23. Component-based software engineering
24. Computer-supported cooperative work
25. Configuration management and deployment
26. Crowd sourced software engineering
27. Cyber physical systems
28. Data-driven software engineering
29. Debugging
30. Dependability, safety, and reliability

List of Course Group Discussion Topics:

16. Distributed and collaborative software engineering
17. Domain modelling and meta-modelling
18. Education
19. Embedded software
20. Emerging domains of software
21. Empirical software engineering
22. End-user software engineering
23. Fault localization
24. Formal methods
25. Green and sustainable technologies
26. Human and social aspects of software engineering
27. Human-computer interaction
28. Knowledge acquisition and management
29. Machine learning for software engineering
30. Middleware, frameworks, and API

List of Home Assignments:

Design:

7. Software visualization

8. Specification and modeling languages
9. Tools and environments
10. Traceability
11. Ubiquitous and pervasive software systems
12. Validation and verification

Case Study:

7. Software economics and metrics
8. Software engineering for machine learning
9. Software evolution and maintenance
10. Software modeling and design
11. Software process
12. Software product lines

Blog

13. Mining software engineering repositories
14. Mobile applications
15. Model-driven engineering
16. Parallel, distributed, and concurrent systems
17. Performance
18. Program analysis
19. Program comprehension
20. Program repair
21. Program synthesis
22. Programming languages
23. Recommendation systems
24. Refactoring

Surveys

11. Requirements engineering
12. Reverse engineering
13. Safety-critical systems
14. Scientific computing
15. Search-based software engineering
16. Security, privacy and trust
17. Software architecture
18. Software reuse
19. Software services
20. Software testing

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

MSE+ESE+HA+LAB+CP+CVV+SEMINAR+GD

Text Books: ***(As per IEEE format)***

4. *Ian Sommerville, 'Software Engineering', Addison-Wesley, 9th Edition, 2010, ISBN-13: 978-0137035151.*
5. *Kenneth S. Rubin, Essential SCRUM: A Practical Guide To The Most Popular Agile Process, Addison-Wesley, ISBN-13: 978-0-13-704329-3, 2012*
6. *Tom Pender, "UML Bible", John Wiley & sons, ISBN – 0764526049*

Reference Books: ***(As per IEEE format)***

7. *SorenLauesen, Software requirements: Styles and techniques, Addison Wesley, ISBN 0201745704, 2002*
8. *Dean Leffingwell, Agile Software Requirements, Addison-Wesley, ISBN-13: 978-0-321-63584-6, 2011*
9. *Charles G. Cobb, The Project Manager's Guide To Mastering Agile: Principles and Practices for an Adaptive Approach, Wiley Publications, ISBN: 978-1-118-99104-6 (paperback), ISBN 978-1-118-99177-0 (epdf), 2015*
10. *Grady Booch, James Rambaugh, Ivar Jacobson, "Unified Modeling Language Users Guide", 2nd Edition, Addison- Wesley, ISBN – 0321267974*
11. *Erich Gamma, Richard Helm, Ralph Johnson, "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley Professional, ISBN-10: 0201633612 ISBN-13: 978-0201633610*
12. *Paul Clements, Felix Bachmann, Len Bass, David Garlan, Documenting Software Architectures: Views and Beyond Addison-Wesley Professional 2003, ISBN-10:0201703726, ISBN-13: 9780201703726*

Moocs Links and additional reading material: www.nptelvideos.in

www.nptelvideos.in
www.coursera.com

www.udemy.com

Course Outcomes:

7. Summarize capabilities and impact of Software Development Process Models and justify process maturity through application of Software Engineering principles and practices focusing tailored processes that best fit the technical and market demands of a modern software project.
8. Discriminate competing and feasible system requirements indicating correct real world problem scope and prepare stepwise system conceptual model using stakeholder analysis and requirement validation.
9. Formulate system specifications by analyzing User-level tasks and compose software artifacts using agile principles, practices and Scrum framework along with Propose and demonstrate realistic solutions supported by well-formed documentation with application of agile roles, sprint management, and agile architecture focusing project backlogs and velocity monitoring.
10. Compose system analysis and design specifications indicating logical, physical, deployment, and concurrency viewpoints using object-oriented analysis and design principles and Model Driven Engineering practices using UML-supported modeling tools.
11. Comprehend the nature of design patterns by understanding a small number of examples from different pattern categories and apply these patterns in creating a correct design using design heuristics, published guidance, applicability, reasonableness, and relation to other design criteria resulting in well-documented system profiles to the engineering and social community.
12. Propose multi-faceted defendable solutions demonstrating team-skills accommodating design patterns reducing the potential cost and performance impedance in order to realize system artifacts with the help of Model Driven Development practices using, scheduling, estimation and risk management activities.

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO3	PO4	PO8	PO11	PSO3
2	3	3	2	1	3

CO attainment levels

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO3	PO4	PO8	PO11	PSO3
1	5	2	3	3	4

Future Courses Mapping:

Software testing and Quality Assurance, Service-oriented Software

Job Mapping:

Application Architect, Project Designer, SCRUM Role Players

FF No. : 654

CS3202 :: ARTIFICIAL INTELLIGENCE

Course Prerequisites: Mathematics, Data Structures

Course Objectives:

1 To introduce the concepts, techniques and building blocks of Artificial Intelligence.

2 To evaluate the searching Techniques and its implementation

3 To analyze planning techniques and its applications in developing solutions to real-world problems.

4 To apply information gained through knowledge representation and uncertainty to a given situation

5 To generate an ability to design, analyze and perform experiments on real life problems using various AI Techniques.

6 To build artificial intelligence models and implement in real life scenario.

Credits:.5..... **Teaching Scheme Theory:…3 Hours/Week**

Tut: 1 Hours/Week

Lab:...2.Hours/Week

Course Relevance: Artificial intelligence (AI) is currently one of the hottest buzzwords in tech and with good reason. The last few years have seen several techniques that have previously been in the realm of science fiction slowly transformed into reality. The importance of Artificial Intelligence has been increasing as a growing number of companies are using these technologies to improve their products and services, evaluate their business models, and enhance their decision-making process.

SECTION-1

Topics and Contents

Fundamentals of Artificial Intelligence: Introduction, A.I. Representation, Non-AI and AI Techniques, State Space Search, Production Systems, Problem Characteristics, types of production systems, Intelligent Agents and Environments, concept of rationality, structure of agents, problem solving agents, and problem formulation.

Searching: Depth First Search, Breadth First Search, Generate and test, Hill Climbing, Best First Search, A* and AO* Algorithm, Constraint satisfaction, Means-Ends Analysis. Game playing: Minimax Search, Alpha-Beta Cutoffs.

Planning: Blocks world, STRIPS, Implementation using goal stack, Partial Order Planning, Hierarchical planning, and least commitment strategy. Conditional Planning, Continuous Planning.

SECTION-1I

Topics and Contents

Knowledge Representation: Knowledge based agents, Wumpus world, Propositional Logic: Representation, Inference, Reasoning Patterns, Resolution, First order Logic: Representation, Reasoning Patterns, Forward and Backward Chaining. Basics of PROLOG: Representation, Structure, Backtracking, Expert System, MYCIN, DART, ZOON,

Uncertainty: Non Monotonic Reasoning, Logics for Non Monotonic Reasoning, Forward rules and Backward rules, Justification based Truth Maintenance Systems, Semantic Nets Statistical Reasoning, Markov Networks.

List of Tutorials: (Any Three)

1) Intelligent Agents and Environments

2) Uninformed searching Techniques

3) Informed searching Techniques

4) Gaming

5) Planning

6) Blocks world

7) STRIPS

7) Wumpus world

8) Propositional Logic

9) Expert System

10) Chat bots and Robotics

List of Practical's: (Any Six)

1. Tic-Tac-Toe Game playing
2. Analysis of AI and Non-AI technique by implementing any two player game.
3. Implement Uninformed (Exhaustive) searching Technique/s
4. Implement Informed (Heuristic) searching Technique/s
5. Implement steepest ascent hill climbing for 8-puzzle/other application
6. Wumpus world
7. Propositional Logic
8. Expert System

9. Chat bots and Robotics
10. Implement simple hill climbing for 8-puzzle / other application
11. Analysis of Constraint satisfaction Problems.
12. Implement Robotic hand
13. Implement Perception model for detection

List of Projects:

1 Inventory management E Commerce

2 stock market price prediction

3 Identification / detection

4 Product Delivery Drones

5 Pick and drop robotic arm

6 Arrangement of blocks

7 Smart city water / light management system

8 Human Tracking system

9 Automatic Interview Conduction system

10 Student Information Chatbot Project.

11. Product Review Analysis For Genuine Rating.

12 Customer Targeted E-Commerce

13 College Enquiry Chat Bot

14 Artificial Intelligence HealthCare Chatbot System

15 Intelligent Tourist System Project

List of Course Seminar Topics:

1) Basic Concepts Artificial Intelligence

2) Intelligent Agents

3)Uninformed searching Techniques

3) Informed searching Techniques

4) Gaming Techniques

5) Planning Techniques

6) Applications of AI

7) Wumpus world

8) Propositional Logic

9) Expert System

10) Chat bots

11) AI Robots

List of Course Group Discussion Topics:

1. Artificial Intelligence and Machine Learning

2. Artificial Intelligence and Data science

3. Artificial Intelligence applications

4. Artificial Intelligence future

5. Artificial Intelligence after 10 years / 2030

6. Uninformed searching and Informed searching Techniques

7. Chatbots and Recommender systems

8. Will Automation and AI Reduce or Increase Jobs.

9. Cashless Economy using AI

10. AI in covid-19 situations

List of Home Assignments:

Design:

1. AI Accessibility Design
2. Best Artificial Intelligence Design for any one application
3. AI Customer Experience Design
4. AI Data-Informed Design
5. AI Decision Architecture
6. AI application Designing for Children
7. AI Designing for Senior Citizens
8. AI in eCommerce Design
9. AI Enterprise UX Design
10. AI Experience Design

Case Study:

1. How Mercedes Is Preparing For The 4th Industrial Revolution using AI

2. How Indian Retail Giant Is Using AI And Robots To Prepare For The 4th Industrial Revolution

3. Rolls-Royce And Google Partner To Create Smarter, Autonomous Ships Based On AI

4. The Amazing Ways Tesla Is Using Artificial Intelligence And Big Data

5. The Incredible Ways John Deere Is Using Artificial Intelligence To Transform Farming

Blog

1. AI Trends

2. AI Research

3. AI Chatbot

4. Chatbot Magazine

5. AI Medical / Agriculture

Surveys

1. Adaption of AI in 2020

2. AI in Industry

3. AI in Digital Marketing

4. AI in Gaming

5. AI after Covid-19

Suggest an assessment Scheme:

HA, PPT,GD,MSE,ESE,LAB,CVV

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

Text Books: (As per IEEE format)

1.*Elaine Rich and Kevin Knight: "Artificial Intelligence." Tata McGraw Hill*

2. Stuart Russell & Peter Norvig : "Artificial Intelligence : A Modern Approach", Pearson Education, 2nd Edition.

3. Deepak Khemani: "A First Course in Artificial Intelligence", Mc Graw Hill

4. Saroj Kaushik: "Artificial Intelligence" Cengage Publication

Reference Books: ***(As per IEEE format)***

1. Ivan Bratko : "Prolog Programming For Artificial Intelligence" , 2nd Edition Addison Wesley,1990.

2. Eugene, Charniak, Drew Mcdermott: "Introduction to Artificial Intelligence.", Addison Wesley

3. Patterson: "Introduction to AI and Expert Systems", PHI

4. Nilsson : "Principles of Artificial Intelligence", Morgan Kaufmann.

5. Carl Townsend, "Introduction to turbo Prolog", Paperback, 1987

Moocs Links and additional reading material: ***www.nptelvideos.in***

www.eduplus.in

Course Outcomes:

Upon completion of the course, graduates will be able to –

1. Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.

2. Formalize a given problem in the language/framework of different AI methods (e.g., as a search problem, as a constraint satisfaction problem, as a planning problem, as a Markov decision process, etc)

3. Implement basic AI algorithms (e.g., standard search algorithms or dynamic programming).

4. Design and carry out an empirical evaluation of different algorithms on a problem formalization, and state the conclusions that the evaluation supports.

5. Use various symbolic knowledge representations to specify domains and reasoning tasks of a situated software agent.

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO4	PO5	PO6	PO10	PSO3
2	3	3	2	1	3

CO attainment levels

CO1-3 CO2-3 CO3-1 CO4-2 CO5-2 CO6-2

Future Courses Mapping:

Artificial Neural Networks, Machine Learning, Soft Computing, Deep Learning.

Job Mapping:

AI Scientist, AI Developer, AI Designer,

FF No. : 654

CS3207::Compiler Design

Course Prerequisites: Automata Theory (grammar)

Course Objectives:

1. Understand the process of program execution cycle.
2. Understand the translation process from High Level Languages to Machine Level Language.
3. Know the syntax and semantic analysis approaches for efficient code/program verification.
4. Learn the methods of code generation which helps for the optimization.
5. Learn code optimization and runtime code synthesis.
6. Know the process of compiler design for emerging programming languages.

Credits:.4..... **Teaching Scheme Theory:…3 Hours/Week**

Tut: Hours/Week

Lab:...2.Hours/Week

Course Relevance: All high level programming languages are easy for users to understand but not understood by a computing machine. The computing machine knows only binary data. A translation is required, in this case, to convert higher level language into machine level, so that the intended program could execute. This translation is done by using a compiler. This course will give you detailed insights of how compilers function internally and design it efficiently. This gives freedom to design your own programming language with it's compiler.

SECTION-1

Topics and Contents

Compilers: Introduction to compiler phases, introduction to cross compiler, features of machine-dependent and independent compilers, overview of types of compilers. Interpreters: compiler vs. interpreter, phases, and working, Preprocessor: header file and macro expansion.
Assembler: Elements of assembly language programming, design of the assembler, assembler design criteria, types of assemblers, two-pass assemblers, one-pass assemblers, assembler algorithms, multi-pass assemblers, variants of assemblers design of two-pass assembler, machine-dependent and machine-independent assembler features.

Linkers: Relocation and linking concepts, static and dynamic linker, subroutine linkages.
Loaders: Introduction to the loader, loader schemes: compile and go, general loader scheme, absolute loaders, relocating loaders, direct linking loaders, MS DOS linker.
Lexical Analysis and introduction to Syntax Analysis: Introduction to Compiler, Phases and Passes, Bootstrapping, Role of a Lexical Analyzer, Specification and Recognition of Tokens, LEX/FLEX, Expressing Syntax, Top-Down Parsing, Predictive Parsers. Implementing Scanners, operator precedence parsers.
Syntax and Semantic Analysis: Bottom-Up Parsing, LR Parsers: constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, YACC/BISON Type Checking, Type Conversion. Symbol Table Structure.

SECTION-II

Topics and Contents

Syntax-Directed Translation and Intermediate Code Generation: Syntax-Directed Definitions, Bottom-Up Evaluation, Top-Down Translation, Intermediate Representations, Intermediate Code Generation. Error Detection & Recovery: Lexical Phase errors, syntactic phase errors, semantic errors. More about translation: Array references in arithmetic expressions, case statements, introduction to SSA form
Code Generation: Issues in Code Generation, Basic Blocks and Flow Graphs, Next-use information, A simple Code generator, DAG representation of Basic Blocks, Peephole Optimization. Generating code from dags.
Code Optimization and Run-Time Environments: Introduction, Principle Sources of Optimization, Optimization of basic Blocks, Introduction to Global Data Flow Analysis, Runtime Environments, Source Language issues. Storage Organization, Storage Allocation strategies, Access to non-local names, Parameter Passing
Machine Dependant Optimization: Instruction (Basic-Block) scheduling algorithm, Instruction selection algorithm, Register allocation techniques, peephole optimizations
Introduction to Data flow analysis: Introduction to constant propagation, live range analysis
Case studies: LLVM compiler Infrastructure, Power of SSA, compiling OOP features, Compiling in multicore environment, Deep learning compilation,

List of Practical's: (Any Six)

1) LEX/FLEX specification and programming regular expressions

2) Add line numbers to lines of text, printing the new text to the standard output using LEX/FLEX.

3) Implement LEX/FLEX code to select only lines that begin or end with the letter 'a' and delete everything else.

4) Implement LEX/FLEX code to count the number of characters, words and lines in an input file.

5) Implement LR/SLR/LALR Parser.

6) Implement Syntax directed Translator.

5) Convert all uppercase characters to lowercase except inside comments.

6) Change all numbers from decimal to hexadecimal notation, printing a summary

statistic (number of replacements) to stderr.

7) Implement Lexical Analyzer for language C-.

8) YAAC specifications and implement Parser for specified grammar.

9) Implement Parser for language C-.

10) Implement an Intermediate code generator (three address code and Quadruples)

List of Projects:

1. Compiler for subset of C using Lex and YAAC

2. Compiler for Subset of Java programming Language

3. Intermediate Code generator

4. Code Optimizer

5. Develop an Editor for Assembly programming. (Use available Assembler MASM/TASM to compile the code and execute in editor)

6. Design a system to check syntax and semantics of English Language.

7. Design a system to check syntax and semantics of a subset of Logical programming Language.

8. Design a System to check syntax and semantics of a subset of Python programming language.

9. Compiler for subset of C++ programming language

10. Compiler for a subset of Algol programming language

List of Course Seminar Topics:

1. Tools complementary to Lex
2. Tools complementary to YAAC
3. Semantic Analyser
4. Obsolete programming Language compiler advantage and issues
5. Android App program compiler
6. Approaches of Intermediate Code generation
7. Recent Trends in Compiler
8. Recent Trends in Interpreter
9. Decompilation
10. Compilation in multicore machines

List of Course Group Discussion Topics:

1. Compiler Vs Interpreter
2. Multi Language Compiler
3. Tree structure for parsing
4. Decompilers: Good or Bad
5. Universal Compiler
6. Cross compiler
7. Alternate to parsers
8. Compiler challenges in mobile app development.
9. Online Compilers

10. Compilers in field of Game development

List of Home Assignments:

Design:

1. Recent methodologies in Intermediate Code Generator

2. Recent methodologies in Code Optimizer

3. Universal Compiler

4. Compiler for Deep learning

5. Recent trend in parsers

Case Study:

1. Algol Compiler

2. Compilation process(internals) of Functional Programming

3. Compilers for Mobile App development

4. LLVM compiler

5. Cross compiler

Blog

1. Decompilers: Ethical or Unethical?

2. Multiparadigm programming compiler

3. State of the Art tools for rapid compiler development

4. Compiler for parallel machines

5. Compiler for distributed computing

Surveys

1. Obsolete Programming Language Compilers

2. Obsolete Programming Language Interpreter

3. Compilers for various programming paradigms

4. Online compilers

5. Mobile app cross compiler

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360-degree assessment and check if it covers all aspects of Bloom's Taxonomy.

MSE(15)+ESE(15)+HA(10)+LAB(10)+CP(10)+CVV(20)+SEMINAR(10)+GD(10)

Text Books: ***(As per IEEE format)***

1. *Aho, A.V., Lam, M.S., Sethi, R., & Ullman, J.D. (2006). Compilers: Principles, Techniques, and Tools, Addison Wesley, ISBN 978-81317-2101-8 (2nd Edition).*

2. *Cooper, K., & Torczon, L. (2011). Engineering a compiler. Morgan Kaufmann, ISBN 155860-698-X.*

3. Appel, A. W. (2004). Modern compiler implementation in C. Cambridge university press.

4. Appel, A. W., & Jens, P. (2002). Modern compiler implementation in Java. In ISBN 0–521–58388–8. Cambridge University Press.

5. Appel, A. W. (1998). Modern Compiler Implementation in ML, In ISBN 0-521-60764-7. Cambridge University Press.

6. Raghavan, V. (2010). Principles of Compiler Design. Tata McGraw-Hill Education.

Reference Books: ***(As per IEEE format)***

1. *Muchnick, S. (1997). Advanced compiler design implementation. Morgan Kaufmann, ISBN 8178672413*

2. *Levine, J. R., Mason, J., Levine, J. R., Mason, T., Brown, D., Levine, J. R., & Levine, P. (1992). Lex & yacc. "O'Reilly Media, Inc".*

Moocs Links and additional reading material: www.nptelvideos.in

https://swayam.gov.in/nd1_noc20_cs13/preview

https://www.udacity.com/course/compilers-theory-and-practice--ud168

https://online.stanford.edu/courses/soe-ycscs1-compilers

Course Outcomes:

1) Design basic components of a compiler including scanner, parser, and code generator.

2) Perform semantic analysis in a syntax-directed fashion using attributed definitions.

3) Apply local and global code optimization techniques.

4) Synthesize machine code for the runtime environment.

5) Develop software solutions for the problems related to compiler construction.

6) Adapt themselves to the emerging trends in language processing.

CO PO Map

CO1-PO2 - 2

CO2-PO3 - 3

CO3-PO4 - 3

CO5-PO11 - 2

CO6-PO12 - 1

CO4-PSO3 – 3

CO attainment levels

CO1 – 2, CO2 – 3, CO3 – 3, CO5 – 3, CO4 – 4, CO6 - 5

Future Courses Mapping:

Mention other courses that can be taken after completion of this course

Job Mapping:

Software Engineer, Compiler Developer

Syllabus Template

CS3224::Engineering Design and Innovation-VI

Course Prerequisites: Problem Based Learning

Course Objectives:

1. To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problems.
2. To Evaluate alternative approaches, and justify the use of selected tools and methods,
3. To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
4. To engage students in rich and authentic learning experiences.
5. To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
6. To develop an ecosystem to promote entrepreneurship and research culture among the students.

Credits:.4..... **Teaching Scheme Theory:… Hours/Week**

Tut: Hours/Week

Lab:.8.Hours/Week

Course Relevance: Project Centric Learning (PCL) is a powerful tool for students to work in areas of their choice and strengths. Along with course based projects, curriculum can be enriched with semester long Engineering Design and Development courses, in which students can solve socially relevant problems using various technologies from relevant disciplines. The various socially relevant domains can be like Health care, Agriculture, Defense, Education, Smart City, Smart Energy and Swaccha Bharat Abhiyan. To gain the necessary skills to tackle such projects, students can select relevant online courses and acquire skills from numerous sources under guidance of faculty and enrich their knowledge in the project domain, thereby achieving project centric learning. Modern world sustained and advanced through the successful completion of projects. In short, if students are prepared for success in life, we need to prepare them for a project-based world. It is a style of active learning and inquiry-based learning. Project based learning will also redefine the role of teacher as mentor in the learning process. The PCL model focuses the student on a big open-ended question, challenge, or problem to research and respond to and/or solve. It brings students not only to know, understand and remember rather it takes them to analyze, design and apply categories of Bloom's Taxonomy.

Preamble - The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith. For all courses of ED, laboratory course contents of "Trends in Engineering Technology" are designed as a ladder to extend connectivity of software technologies to solve real word problem using interdisciplinary approach. The ladder in the form of gradual steps can be seen as below:

Industry Communication Standards, Single Board Computers and IoT, Computational Biology (Biomedical and Bioinformatics), Robotics and Drone, Industry 4.0 (Artificial Intelligence, Human Computer Interfacing, 5G and IoT, Cloud Computing, Big Data and Cyber Security etc).

Group Structure:

- There should be a team/group of 4-5 students.
- A supervisor/mentor teacher assigned to individual groups.

· It is useful to group students of different abilities and nationalities together.

Selection of Project/Problem:

- Students must focus to initiate the task/idea .The idea inception and consideration shall be from following areas as a real world problem:
- Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
- This is the sample list to start with. Faculty and students are free to include other areas which meet the society requirements at large.
- The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher's Role in PCL :

- Teacher is not the source of solutions rather he will they act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Student's Role in PCL:

- Students must have ability to initiate the task/idea .they should not be mere imitators.
- They must learn to think.
- Students working in PCL must be responsible for their own learning.
- Students must quickly learn how to manage their own learning, Instead of passively receiving instruction.
- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution. How effective is …….? How strong is the evidence for ……? How clear is ……?
- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning

- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively.
- Skills required by students in information literacy include:
- How to prepare the search? How to carry out the research
- Sorting and assessing of information in general

Use of Research Methodology: - investigation, collaboration, comprehension, application, analysis, synthesize and evaluation

Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

EDI Sample Case Studies : -

1) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis

2) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis for SAM and BAM processing and analysis

3) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis for Gnome NGS processing and analysis

4) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis for population genetics simulation

5) Design of bioinformatics library using CPython for Next-generation Sequencing processing and analysis for proteomics processing and analysis

...not limited to.....Faculty and students are free to include other area which meets the society requirements at large.

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Bloom's Taxonomy.

Higher levels of the Booms Taxonomy - analyze, apply, evaluate and create.

Text Books: ***(As per IEEE format)***

1. A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017

2. Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.

Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Robart Capraro, Mary Margaret Capraro

Reference Books: ***(As per IEEE format)***

1. De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based learning in engineering. Rotterdam: Sense Publishers. 2007.

2. Project management core textbook, second edition, Indian Edition , by Gopalan.

3. The Art of Agile Development. By James Shore & Shane Warden.

Course Outcomes:

On completion of the course, learner will be able to–

CO1: Identify the real life problem from societal need point of view

CO2: Choose and compare alternative approaches to select most feasible one

CO3: Analyze and synthesize the identified problem from technological perspective

CO4: Design the reliable and scalable solution to meet challenges

CO5: Evaluate the solution based on the criteria specified

CO6: Inculcate long life learning attitude towards the societal problems

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO1	PO2, PO4	PO3, PO5	PO4, PO5, PO8	PO6, PO7, PO9, PO11	PO12, PSO2, PSO3, PSO4
3	3, 3	2, 3	2, 2, 2	2, 2, 2, 1	2,3,2,2

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

Job Mapping:

What are the Job opportunities that one can get after learning this course

Software Engineer, Software Developer, IT Engineer

Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Structure & Syllabus of

Final Year B. Tech.
(Computer Engineering)

Pattern "D21"

Effective from Academic Year 2021-22

Prepared by: - Board of Studies in Computer Engineering

Module VII Courses

FF No. : 654

MD4205: Marketing Management

Credits: 02 **Teaching Scheme: - Theory 2 Hrs/Week Lab: 0 hrs /week**

Section-I

1. **Concepts of Marketing**
   Definition of Marketing, Core marketing concepts, Marketing Management philosophies, Micro and Macro Environment, Characteristics affecting Consumer behavior, Types of buying decisions, buying decision process, Classification of consumer products, Market Segmentation
2. **Marketing Information Systems And Research**
   Components of marketing information system–benefits & uses marketing research system, marketing research procedure, Demand Estimation research, Test marketing, Segmentation Research - Cluster analysis, Discriminate analysis. Sales forecasting: objective and subjective methods
3. **Marketing Of Industrial Goods**
   Nature and importance of the Industrial market, classification of industrial products, participants in the industrial buying process, major factors influencing industrial buying behavior, characteristics of industrial market demand. Determinants of industrial market demand Buying power of Industrial users, buying motives of Industrials users, the industrial buying process, buying patterns of industrial users.

Section -II

1. **Product Management**
   The concept of a product, features of a product, classification of products, product policies – product planning and development, product line, product mix – factors influencing change in product mix, product mix strategies, meaning of "New – product; major stages in new – product development product life cycle.
2. **Branding**
   Reasons for branding, functions of branding features of types of brands, kinds of brand name.
3. **Pricing Policies**
   Importance of Price, pricing objectives, factors affecting pricing decisions, procedure for price determination, kinds of pricing, pricing strategies and decisions,
4. **Advertising and Sales Promotion**
   Objectives of advertisement function of advertising, classification of advertisement copy, advertisement media – kinds of media, advantages of advertising. Objectives of sales promotion, advantages sales promotion,
5. **Packaging**
   Meaning, growth of packaging, function of packaging, kinds of packaging.

List of Practical: -

Student will do exercises or case studies based on following topic

1. Analysis of Factors affecting Consumer behavior
2. Identification of variables for market segmentation
3. Components of marketing information system
4. Cluster analysis for Market Research
5. Questionnaire design for collecting primary data for Market Research
6. Case study on Marketing of Industrial Goods
7. Case study on Product Line Management
8. Product life cycle analysis
9. Case study on Product Promotion strategies
10. Case study on Pricing policies
11. Case study on Labeling & Packaging
12. Case study on Branding

List of Projects

Students will perform following projects

1. Consumer Behavior Analysis
2. Market Segmentation Analysis
3. Exploratory Research for Market Competition Analysis
4. STP (Segmentation, Target, Positioning) Analysis
5. Analysis of B2B Marketing
6. Forecasting for Market Analysis
7. Designing Marketing Information System
8. Designing Product Promotion Mix
9. Pricing Policy Impact Analysis
10. Data collection & analysis for Market Research
11. Cluster Analysis for Market Segmentation
12. Market Analysis for New product development

List of GD Topics

1. Advertisements- helpful to customers or just eye wash
2. Advertising is all glitter and no substance.
3. Consumer is never satisfied.
4. Consumer is the king in today's market.
5. Commitment is more important than other skills in marketing
6. Digital marketing via Blogs versus Video - Which is more effective?
7. Should a start-up invest heavily in Marketing
8. Quality is the key to successful Marketing
9. A career in marketing - Worth it or not?
10. Lying for sale of products should be avoided in Marketing

Text Books: ***(As per IEEE format)***

1.	Philip Kotler,	Principles of Marketing	Prentice – Hall.
2.	Philip Kotler,	Marketing Management	Prentice – Hall.

Reference Books: ***(As per IEEE format)***

1.	Wiliam J Stanton	Fundamentals of Marketing	McGraw Hill
2.	R.S.N. Pillai and Mrs. Bagavathi	Marketing	S. Chand & Co. Ltd

Course Outcomes:

Students will be able to:

1. Understand basic marketing management concepts and their relevance to business development
2. Prepare a questionnaire for market research
3. Understand consumer & industrial buying decision process & motives.
4. Understand the concept of product management and branding in context of consumer and industrial products
5. Design marketing research plan for business organizations.
6. Optimize marketing mix to get competitive advantage

FF No. : 654

Syllabus Template

CS4217:: Human-Computer Interaction

Course Prerequisites: NA

Course Objectives:

1. To categorize IT applications based on measurable human factors,

2. To study the user community through user survey and/or field visit,

3. To design user-friendly user interfaces with due consideration of interface theory and principles,

4. To apply usability evaluation methods to identify the usability issues with IT applications,

5. To understand the kind of documentation required for IT applications,

6. To integrate web and mobile app design approaches as per user requirements

Credits: 02 **Teaching Scheme Theory: 02 Hours/Week**

Course Relevance: Human-Computer Interaction (HCI) is a multi-disciplinary socio-technical course, with a goal of bringing the power of computers and communication systems to users, customers or people. It aims to make all computing and communications systems more accessible, maintainable and useful in working, learning and recreational life of their users. It helps every computing, web or mobile application to become really user-centered, increasing its users as well as related sales.

SECTION-1

Topics and Contents:

Fundamentals of Human Computer Interaction (HCI): Definition of HCI, Interdisciplinary Nature, Related Disciplines, Usability, Types of Usability, User Interface (UI), Measurable Human Factors, Accessibility, Differently-abled Users, Accessibility Guidelines.

Interaction Concepts and Models: User Persona, User Categorization, Golden Rules of Interface Design, Miller's Principle, Task Analysis - GOMS, Contextual Inquiry, Work Models, Interaction Styles, Empathy Maps.

Design Process: Design Concept, Three Pillars of Design, Process of Design, Ethnographic Observations, Participatory Design, Internationalization, Interaction Design Patterns.

SECTION-1I

Topics and Contents:

Usability Evaluation: Expert-based Evaluation, User-based Evaluation, Formative Evaluation, Summative Evaluation, Heuristic Evaluation, Cognitive Walkthrough, Semiotic Analysis, Icon Categorization, User Surveys, Interviews, Usability Testing.

Documentation and Groupware: Classification of Documents, Reading from Displays, Online Help, Tutorials, Error / Warning Messages, Groupware, Computer Supported Cooperative Work (CSCW), Dimensions of Cooperation, Asynchronous Interactions, Synchronous Interactions, Challenges with Online Communications.

Website and Mobile App Design: Content Design, Interaction and Navigation Design, Presentation Design, Differences in design approaches, Design and Evaluation Tools.

List of Home Assignments:

Designs:

1. Social Network for Spiritual Users
2. App for Alzheimer's disease
3. Health Tracking App
4. Ration Card Management App
5. Innovative e-Commerce Platform

Case Studies:

1. Chatbot in healthcare domain
2. Best food ordering app in India
3. Online teaching-learning process
4. Use of Twitter with Indian Users
5. User experience with car booking in India

Blogs:

1. Noise of Notifications
2. Challenges in Food Delivery Service
3. Need for Accessibility Guidelines
4. Usability of Autonomous Vehicles

5. Failure of Usability Testing

Surveys:

1. User experience with video-conferencing apps
2. User errors on Social Networking Sites (SNS)
3. Challenges for hearing impaired users with IT applications
4. Most popular Indian mobile apps (Made in/by India)
5. Impact of ban on Chinese apps in India

Suggest an Assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

Home Assignments, MSE, ESE

Text Books: ***(As per IEEE format)***

1. *Ben Shneiderman, "Designing the User Interface", Third Edition, Pearson Education, ISBN 81-7808-262-4.*

2. Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, "Human-Computer Interaction", Third Edition, Pearson Education, ISBN 81- 297-0409-9.

Reference Books: ***(As per IEEE format)***

1. *Donald Norman, "The Design of Everyday Things", 2002 Edition, Basic Books, ISBN 100-465-06710-7.*

2. Wilbert Galitz, "The Essential Guide to User Interface Design", Second Edition, Wiley-Dreamtech India (P) Ltd., ISBN 81-265-0280-0.

3. John Carroll, "Human-Computer Interaction in the New Millennium", Pearson Education, ISBN 81-7808-549-6.

Moocs Links and additional Reading Material:

https://nptel.ac.in/courses/106/103/106103115/

https://www.coursera.org/learn/human-computer-interaction

https://classroom.udacity.com/courses/ud400

Course Outcomes:

1. Students will be able to appreciate the differences among IT applications and their categories based on measurable human factors.

2. Students will be able to study the user community through user survey and/or field visit.

3. Students will be able to design user-friendly user interfaces as per user requirements and UI design principles.

4. Students will be able to apply a suitable usability evaluation method to identify the usability issues.

5. Students will be able to understand the kind of documentation required for IT applications.

6. Students will be able to enhance UI designs as per desired web or mobile app design approach.

CO PO Map:

CO1	CO2	CO3	CO4	CO5	CO6
PO3	PO4	PO8	PO10	PO11	PSO4
3	3	2	2	1	3

CO Attainment Levels:

CO1 ----------3

CO2 ----------3

CO3 ----------2

CO4 ----------2

CO5 ----------1

CO6 ----------3

Future Courses Mapping:

User Interface Design,

Usable Security,

Intelligent User Interfaces
Job Mapping: *UI Designer, Product Designer, Software Engineer, Mobile App Developer and other jobs*

FF No. : 654

Syllabus Template

CS4219::Internet of Things

Course Prerequisites: Microprocessor Hardware, Microcontroller

Course Objectives:

1. To learn the terminology, technology and its applications of IoT
2. To analyze Embedded suite widely used in IoT.
3 .To describe the concept of M2M with necessary protocols
4. To understand the cloud storage for IoT applications.
5. To optimize resources for different IoT applications
6. To understand Real world IoT Design constraint.

Credits: 02 **Teaching Scheme Theory: 02 Hours/Week**

Course Relevance:

IoT is regarded as the significant frontier that can improve almost all activities in our lives. Most of the devices, which have not previously been connected to the internet, can be networked and respond the same way as smart devices. Internet of Things and related technologies improve the convenience, comforts and security of our homes and be more efficient and cost effective in the way we consume services such as energy. IOT will help track and monitor a huge range of our own physiological functions.

SECTION-1

Introduction and Application to Internet of Things: Need of IoT, Towards the IoT, Strategic Research and Innovation Directions, Future Internet Technologies, IoT Smart X Application: Smart Cities, Smart Energy & Smart Grid, Smart Mobility & transport, Smart Home, Smart Building & Infrastructure, Smart Factory & Manufacturing, Smart Health, Smart Logistics & Retails.

Embedded Suite for IoT: Introduction to Arduino and Raspberry Pi, Understanding the Arduino and Raspberry Pi board and its Components, recognizing the Input/output, GPIO connectivity.

WIRELESS TECHNOLOGIES supporting IoT:

Protocol Standardization for IoT, Machine to machine (M2M) and WSN Protocols, Basics of

RFID, RFID Protocols , Issues with IoT Standardization, Protocols- IEEE 802.15.4, ZigBee, IPv6 technologies for IOT

SECTION-1I

IoT Networking: Star, Mesh, Tree, and Overview of networking Protocols: TCP/IP, 6LowPan, IoT Devices Application Level Protocol Service parameter in MQTT,

IoT: PRIVACY, SECURITY & GOVERNANCE: Overview of Governance, Privacy and Security Issues, Contribution from FP7 Projects, Security, Privacy and Trust in IoT Platforms for Smart Cities, First Steps towards a Secure Platform, Smart Approach. Data Aggregation for the IoT in Smart Cities

Cloud Analytics for IoT Application:

Introduction to cloud computing, Difference between Cloud Computing and Fog Computing: The Next Evolution of Cloud Computing, Role of Cloud Computing in IoT, Connecting IoT to cloud, Cloud Storage for IoT Challenge in integration of IoT with Cloud

List of Tutorials: (Any Three)

1)Introduction to the Internet of Things

2)How IoT works

3) Features of Internet of Things.

4)IoT Applications

5)Advantages of Internet of Things

6)Representative applications of IoT

7)Architectures for the IoT

8) Layered Architectures for the IoT

9)Relevant MAS concepts and techniques

10)Limitations of MAS concepts and techniques and directions for further research

List of Practical's: (Any Six)

1) LED Blinky
2) Switch
3) ADC
4) PIR, Ultrasonic sensor,IR Sensor, Flame Sensor interface with Microcontroller
5) MQ6 Sensor, Humidity sensor, Raindrop Sensor interface
6) Communication over Wifi through Node MCU
7) Voice Controlled Iot application
8) Serial Communication
9) PWM
10) Bluetooth Controlled application(Interfacing Bluetooth Module (HC-05)
11) Control through Android APP
12) RFID Interface
13) Use IOT Comm Protocol (MQTT)

List of Projects:

1.IoT Based Humidity and Temperature Monitoring Using Arduino Uno

2.IoT Weather Reporting system using Raspberry pi.

3.IoT Connected Healthcare Applications.

4.IoT Based Intelligent Traffic Management System

5.IoT Based Smart Parking System Using RFID

6.Smart Irrigation System Using IoT.

7. Waste and water management using IoT

8. Smart Healthcare Solution using IoT

9. Automatic Herbicides Sprayers

10. Fish Feeder

11. Green Corridor

12. Trusted high-quality elderly care

13. Gesture controlled Iot Application

List of Course Seminar Topics:

1.Authentication of Edge-Device in AWS IoT

2.How to Build a Complete IoT Solution with AWS- An Use Case Approach.

3.Exploring IoT Through a Use Case

4.Security Technologies behind SSL

5.A Look at the AWS IoT Ecosystem

6.Wireless Protocols for Internet of Things

7.Sensor Characteristics.

8.Manufacturing Intelligence

9.IoT in the Manufacturing industry

10.IoT and Machine Learning

11. IoT Communication Protocol selection based on application

12. Security and Privacy issue in internet of things

List of Course Group Discussion Topics:

1. Role of Internet of Things in development of India .

2. Manufacturing industries should make efforts to limit contribution to IoT.

3. Should countries put a ban on IoT for children?

4. Should IoT pay more attention to security rather than just expanding its horizon to the extremes?

5. IoT is the next big thing in technology.

6. IoT poses a huge risk to privacy, if they your system is hacked.

7. IoT is the next big thing for hackers trying to have access to your intimate data.

8. Pros and cons of over-usage of IoT at homes and offices.

9. IoT at battlefields will make life of soldiers safer and easier.

10. IoT will make way for robots to rule over humans one day.

11. IoT devices are making people lazier and obese.

12. IoT needs to be regulated before it goes out of limits and poses serious threat.

13. MQTT vs COAP protocol for IoT applications

List of Home Assignments:

Design:

1.IoT Based Smart Waste Management System for Smart City

2.IoT Based Smart Street Light

3.IoT Based Smart Grid System

4. IoT based Water Quality Management system using Arduino

5. IoT Smart Home automation using Node MC

Case Study:

1.IOT Real Time DashBoard

2.IOT and Block chain

3.IOT Sensor Gateway

4.IOT Operational Analytics

5.Autonomous trucks reduce driver fatigue and improve road safety.

6. Industrial Internet of Things

7. Iot Enabled Next Generation Farming

8. The Tesla IoT Car

9. Role of IoT in Supply chain management

10. Ecosystem for Logistics Industry with IoT

Blog

1.Monitoring environmental conditions to improve safety and prevent environmental accidents

2.Today's hard hats and safety goggles could be tomorrow's sensored vests and work sites. Can an IoT enabled ecosystem prevent accidents from happening in the frst place

3.How can machines tell us when they are about to break down?Using connected technologies to

predict maintenance saves clients . time and money

4.Automation benefit from IoT

5.Farmers used to rely on clouds for rain. Today's digital cloud helps maximize crop yield, optimize seeding, automate harvesting, and more.

6. Security and Privacy issues for Iot application

7. Suitability of MQTT protocols for IoT application

8. Interoperability issues in IoT

9.Toward Industry 4.0 With IoT

10. Iot Applications in Logistics and Supply chain managements

11. Data management for IoT applications.

Surveys

1.The future of IOT Connectivity

2. IoT applications value creation for industry

3.IoT involvement in Software development, selling software for IoT products

4.IOT Platform

5.The internet of things (IoT) represents the Fourth Industrial Revolution

6. Cloud support for Iot Application

7. Interdependencies of BIG data and IoT

8. Iot enabled Smart manufacturing

9. Role of Internet of Things for Electric Vehicle

10. Secure Vehicular Area Network

11. Security attacks on IoT Devices

12. IoT vs. Industrial IoT: What's the difference?

Suggest an assessment Scheme:

As per Institute directives.

Mid Sem Exam, End Sem Exam, Home Assignment, Course Project,Group Discussion

Seminar

Text Books: ***(As per IEEE format)***

1. *Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, Sta matis Karnouskos,David Boyle, "From Machine to the Internet of Things: Introduction to a New Age of Intelligence", 1st Edition, Academic Press, 2014. to the Internet of Things:*

2. Francis daCosta, "Rethinking the Internet of Things: A Scalable Approach to Connecting Everything", 1st Edition, Apress Publications, 2013

3. Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence", 1 st Edition, Academic Press, 2014. (ISBN-13: 978-0124076846)

Reference Books: ***(As per IEEE format)***

1. *Marco Schwartz, "Internet of Things with the Arduino Yun", Packt Publishing, 2014*

2. Daniel Minoli John Wiley & Sons ,,Building the internet of things with ipv6 and mipv6,

The Evolving World of M2M Communications, ISBN: 978-1-118-47347-4

3. Cassimally, Hakim, "Designing the Internet of Things", Wiley Publications, ISBN 10:

111843062X

3. Francis daCosta, "Rethinking the Internet of Things: A Scalable Approach to Connecting Everything", 1 st Edition, Apress Publications, 2013. (ISBN-13: 978- 1430257400)

Moocs Links an additional reading material: www.nptelvideos.in

Course Outcomes:

The student will be able to

1. Realize the revolution of Internet in Mobile Devices, Cloud & Sensor Networks

2. Understand building blocks of Internet of Things and characteristics of Sensors and Communication Devices

3. Implement state of the art architecture in IoT

4. Demonstrate the application of IoT in Industrial Automation and identify Real World Design Constraints.

5. Compare and Contrast the use of Devices, Gateways and Data Management in IoT

6. Implement Security protocols to prevent internal and External damage.

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO3	PO5	PO7	PO9	PO10	PSO3

CO attainment levels

PO3 --3, PO5 ---3, PO7 –2, PO9 ---2, PO10---1, PSO3---3

Future Courses Mapping:

Mention other courses that can be taken after completion of this course

Job Mapping:

1) All sectors of business from agriculture, transportation, health, manufacturing, resources, mining and retail to service sector are set to benefit from or at least be impacted by IoT.
2) IOT will offer Local governments a great opportunity to provide a range of services that will not only improve the quality of life of its citizens but also improve the efficiency and profitability of state services.

FF No. : 654

Syllabus Template

CS4222 :: Image Processing

Course Prerequisites: Digital Signal Processing

Course Objectives:

1. To describe different color models and image processing techniques
2. To analyze image condition and deduce enhancement algorithms
3. To understand Segment the image to identify the region of interest.
4. To develop an algorithm to recognize the specified objects in the given image.
5. To learn different compression techniques

Credits: 02 **Teaching Scheme Theory: 02 Hours/Week**

Course Relevance: Vision sense is the most powerful human sense organ. In the world where intelligent automation is taking place, image processing is a vital domain for research and development. In Industry 4.0, image processing systems built around industrial cameras are an essential component in automated production. Throughout all steps of production, from the inspection of raw materials and production monitoring (i.e. flaw detection) to final inspections and quality assurance, they are an indispensable part of achieving high efficiency and quality standards. In the Entertainment Industry, latest trends such as 4K video streaming requires high quality compression that can provide limited/no loss image quality with high fps. In social networking, sharing images has been a vital part. Creating innovative effects and overall manipulating the images will be explored.

SECTION-1

Topics and Contents

Introduction: Elements of image processing system, Scenes and Images, Vector Algebra, Human Visual System, color vision color model: RGB, HVS, YUV, CMYK, YCbCr and some basic relationships between pixels, linear and nonlinear operations, Image sampling and quantization.

Image Enhancements: Memory-less operations, Spatial domain image enhancements: Denoising filters, Smoothing Operation, Sharpening Operation, and Contrast stretching /enhancement, histogram and histogram equalization.

Image segmentation: Classification of image segmentation techniques: Edge-based Segmentation, Region based techniques. Binarization: Global Thresholding, Adaptive thresholding. Types of Edge detector: derivative filters, Sobel, Canny. Edge linking. Feature

Extraction-

SECTION-1I

Topics and Contents

Morphological Operation: Binary Morphology, Erosion Dilation, Opening and Closing.

Feature Extraction and Object Recognition: Feature points and feature detection (Line, circle and corner). Line detection: RANSAC, Hough Transform. Corner detection: Harris Corner Detector. Feature descriptors, Descriptor matching. SIFT, Boundary representation (Chain code), Boundary detection based techniques.

Image compression: Introduction and need, Coding redundancy, classification of compression techniques (Lossy and lossless- JPEG, Run Length Coding, Huffman Coding, Shannon fano coding).

List of Course Seminar Topics:

1. Challenges in Automated Video Surveillance
2. Tumor detection in MRI images
3. Eye gaze tracking for HMI: Pros, cons and implementation
4. Roll of image processing in Industry 4.0
5. Parallelism for performance enhancement in image processing
6. Vision based ADAS
7. Computational photography
8. Computational microscopy
9. Automatic navigation using Visual SLAM
10. Animoji

List of Course Group Discussion Topics:

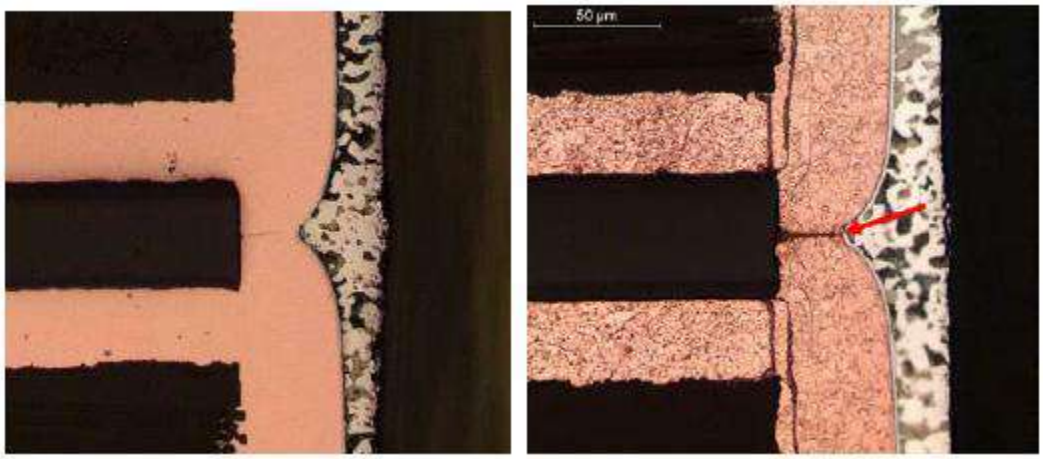
1. Lines Vs. Corners as features
2. Hough Transform for line detection Vs. RANSAC

3. Fourier domain denoising Vs. Spetial domain denoising
4. Kernel size Vs. Speed of operation
5. Histogram equalization Vs. Gamma correction
6. OTSU Vs Adaptive thresholding
7. Compression techniques
8. Color models
9. SIFT Vs SURF
10. Roll of image processing in security

List of Home Assignments:

Design:

1. Design an algorithm to identify fault in a "PCB inspection system" as shown below

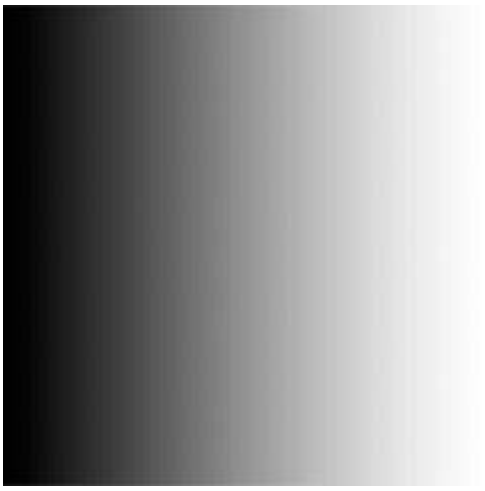


2. Design an algorithm to perform segmentation of the image below to extract the mango from its background

3. Design an algorithm to get from image 1 to image 2

4. Design an algorithm to recognize character "0" in the image below

5. Design an algorithm to compress a 300x300 pixel image with horizontal black to white gradient as shown below



Case Study:

1. Cam-scanner: Document scanning app
2. Tesseract OCR library
3. Instagram filters
4. OpenCV
5. Google Street View

Blog

1. Image processing on Embedded platforms
2. Face recognition system security analysis for authentication
3. Image processing in MSME for effective automation
4. H.264 codec for image streaming
5. Role of mathematics in image processing

Surveys

1. Image quality metrics
2. Vision based self driving car safety
3. Compression techniques & codecs
4. State of the art applications such as AR/ XR
5. Human recognition in social networking apps like Facebook

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

1. HA - 10 Marks
2. Viva 30 Marks
3. Online MCQ-MSE Test 30 Marks
4. Online MCQ-ESE Test 10 Marks

Text Books: ***(As per IEEE format)***

1. *Rafael Gonzalez & Richard Woods, "Digital Image Processing," 3rd Edition, Pearson publications, ISBN 0132345633.*
2. *Anil K. Jain, "Fundamental of Digital Image Processing," 5th Edition, PHI publication, ISBN 13: 9780133361650.*

Reference Books: ***(As per IEEE format)***

1.Pratt, "Digital Image Processing," Wiley Publication, 3rdEdition , ISBN 0-471- 37407-5.

2.K.R. Castleman, "Digital Image Processing," 3rd Edition, Prentice Hall: Upper Saddle River, NJ, 3, ISBN 0-13-211467 -4.

Moocs Links and additional reading material:

1. https://nptel.ac.in/courses/117/105/117105135/
2. https://nptel.ac.in/courses/106/105/106105032/
3. https://nptel.ac.in/noc/courses/noc19/SEM2/noc19-cs58/
4. https://www.coursera.org/learn/computer-vision-basics

Course Outcomes:

The student will be able to

1. Apply various corrective geometric transforms on a distorted image. (PO2)
2. Determine and implement required image enhancement techniques using open source technologies such as OpenCV (PO5)
3. Deploy optimized algorithms for lossless and lossy compression techniques which ensures expected performance on a variety of hardware architectures. (PO8)
4. Contribute to an algorithmic solution for social and personal security (PO10)
5. Differentiate between various mathematical transforms and its use for a given use case (PO12)

6. Deduce a solution for a given industrial problem (PSO4)

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO5	PO8	PO10	PO12	PSO4
3	3	2	2	1	3

CO attainment levels

CO1- 3 CO2- 4 CO3- 3 CO4- 2 CO5-1 CO6-5

Future Courses Mapping:

1. Augmented Reality
2. Multimedia Processing

Job Mapping:

1. Augmented Reality Experience Designer
2. Automation Engineer
3. Embedded Software Developer
4. Image Processing Expert

Syllabus Template

ET4230: NATURAL LANGUAGE PROCESSING

Course Prerequisites:

1. Probability and statistics.
2. Linear Algebra
3. Python programming language

Course Objectives:

1. Learn fundamentals of Text processing
2. Understand the different Language Models
3. Implement POS tagging
4. Implement Text classification
5. Implement sentiment analysis
6. Implement Machine translation

Credits: 2 **Teaching Scheme Theory: 2** Hours/Week

Course Relevance:

Natural Language Processing is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language. The common applications of NLP involves, Google translator, Word Processors such as Microsoft, Interactive Voice Response, Personal assistant applications.

SECTION-1
Text Processing: Basics, Empirical Laws, Spelling Correction: Edit Distance, N-Gram Language Models, Basic Smoothing, POS Tagging, Hidden Markov Models for POS Tagging, Viterbi Decoding for HMM and Parameter Learning, Maximum Entropy Models.
SECTION-1I

Maximum Entropy Models, Name entity recognition, Syntax, Dependency Grammars and Parsing, Semantic, text classification, sentiment analysis, Machine Translation, Question Answering.

List of Course Seminar Topics:

1. SemEval-2016 task 4: Sentiment analysis in Twitter
2. Modelling user attitudes using hierarchical sentiment-topic model
3. Multilingual dynamic topic model
4. Document-Level Text -classification Using Single-Layer Multisize Filters Convolutional Neural Network
5. Twitter Storytelling Generator Using Latent Dirichlet Allocation and Hidden Markov Model POS-TAG (Part-of-Speech Tagging)
7. Part-of-speech Tagging and Named Entity Recognition Using Improved Hidden Markov Model and Bloom Filter
8. Part of speech tagging for Twitter conversations using Conditional Random Fields model
9. A system for named entity recognition based on local grammars
10. A Maximum-Entropy Segmentation Model for Statistical Machine Translation
11. Mobile embodied conversational agent for task specific applications.

List of Course Group Discussion Topics:

1 Smoothing Technique
2. N-gram models
3. POS tagging
4. Ambiguities in NLP
5. Challenges in NLP
6. Challenges in designing Language Translators

7. Challenges in designing text classification
8. Challenges in designing sentiment analysis
9. Challenges in designing Question and Answering system
10. Challenges in designing text summarization.

List of Home Assignments:

Design:

1. POS tagging using HMM
2. Build Chatbot
3. Summarization of customers reviews
4. Social media Information extraction
5. SMS spam classification

Case Study:

1. Hiring and recruitment
2. Advertising
3. Healthcare
4. Market intelligence
5. Sentiment analysis

Blog

1. Social media Information extraction
2. Name Prediction in Multiple Languages using Recurrent Neural Networks
3. Text Classification using Sentiment Analysis
4. Image Caption Generator
5. gender identification in Marathi names

Surveys

1. POS tagging techniques
2. SMS and email spam classification

3. Categorization of sport articles
4. machine translation Techniques
5. Name entity recognition methods

Suggest an assessment Scheme:

Suggest an Assessment scheme that is best suited for the course. Ensure 360 degree assessment and check if it covers all aspects of Blooms Taxonomy.

1. Seminar – 10 Marks
2. Group Discussion – 10 Marks
3. Home Assignment – 10 Marks
4. Course Viva – 20 Marks
5. MSE – 15 Marks
6. ESE – 15 Marks
7. Lab work –10 Marks
8. Course project -10 Marks

Text Books: (As per IEEE format)

1. Jurafsky & Martin "Speech and Language Processing" Prentice Hall, 2000
2. Akshar Bharati, Rajeev Sangal and Vineet Chaitanya: "Natural Language Processing: A Paninian Perspective", Prentice-Hall of India, New Delhi, 1995.

Reference Books: (As per IEEE format)

1. Steven Bird, Ewan Klein, and Edward Loper "Natural Language Processing with Python",
O'Reilly Media 2009.

Moocs Links and additional reading material:

1. https://nptel.ac.in/courses/106/105/106105158/
2. https://nptel.ac.in/courses/106/106/106106211/

Course Outcomes:

The student will be able to –

1. Have broad understanding of the field of natural language processing (Co Attainment Level - 3)
2. Get acquainted with the algorithmic description of the main language levels: morphology, syntax, semantics, and pragmatics (Co Attainment level -3)
3. Apply mathematical models and algorithms in applications of NLP. (Co Attainment level - 4)
4. Design and implementation issues in various NLP applications such as information retrieval and information extraction (Co Attainment level - 4)
5. Demonstrate crucial ideas in linguistics (e.g., syntax, semantics, pragmatics), artificial intelligence (e.g., knowledge representation), and machine learning (e.g., deep learning) to natural language processing. (Co Attainment level - 4)
6. Identify one of the contemporary (sub) problems of natural language processing and implement, in the form of a complete computer program as a possible solution to it.

(Co Attainment level - 5)

CO PO Map

PO	CO 1	CO 2	CO 3	CO 4	CO 5	CO 6	CO 7	CO 8	CO 9	CO 10	CO 11	CO 12	PS O1	PS O2
1	3	3	3	2	4	1		1	1	1		2	1	1
2	3	3	3	2	4	1		1	1	1		2	3	3
3	4	3	3	3	4	2		1	1	1		2	3	3
4	4	3	3	3	4	2		1	1	1		2	3	3
5	4	3	3	2	4	2		1	1	1		2	3	3
6	4	3	3	3	4	2		1	1	1		2	3	3

CO attainment levels

1. **CO1 - Level 3**
2. **C02 - Level 3**
3. **CO3 - Level 4**
4. **CO4 - Level 4**
5. **CO5 - Level 4**
6. **CO6 - Level 5**

Job Mapping:

Natural Language engineers, Data Scientist and Algorithm Architect with industries in domains Media & Entertainment, Healthcare and Finance.

FF No. : 654

Syllabus Template

CS4201::Cloud Computing

Course Prerequisites: Operating Systems, Fundamentals of Computer Networks

Course Objectives:

1. To become familiar with Cloud Computing and its ecosystem
2. To learn basics of virtualization and its importance
3. To evaluate in-depth analysis of Cloud Computing capabilities
4. To give a technical overview of Cloud Programming and Services.
5. To understand security issues in cloud computing

Credits: 2 **Teaching Scheme Theory: 2 Hours/Week**

Course Relevance: Cloud computing to enable transformation, business development and agility in an organization.

SECTION-I

Topics and Contents

Introduction to Cloud Computing: Recent trends in Computing, Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Evolution of cloud computing

Cloud Computing Architecture: Cloud versus traditional architecture, Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS), , Public cloud, Private cloud, Hybrid cloud, Community cloud, Google Cloud architecture, The GCP Console, Understanding projects, Billing in GCP, Install and configure Cloud SDK, Use Cloud Shell, GCP APIs

Infrastructure as a Service (IaaS): Introduction to IaaS, Introduction to virtualization, Different approaches to virtualization, Hypervisors, Machine Image, Virtual Machine(VM), Compute

options in the cloud, Exploring IaaS with Compute Engine, Configuring elastic apps with auto scaling, Storage options in the cloud, Structured and unstructured storage in the cloud, unstructured storage using Cloud Storage, SQL managed services, Exploring Cloud SQL, Cloud Spanner as a managed service, NoSQL managed service options, Cloud Datastore, a NoSQL document store, Cloud Bigtable as a NoSQL option

SECTION-II

Topics and Contents

Platform as a Service (PaaS): Introduction to PaaS, Service Oriented Architecture (SOA). Cloud Platform and Management, Exploring PaaS with App Engine, Event driven programs with Cloud Functions, Containerizing and orchestrating apps with Google Kubernetes Engine

Software as a Service (SaaS): Introduction to SaaS, Web services, Web 2.0, Web OS,

Service Management in Cloud Computing: Service Level Agreements(SLAs), Billing and accounting, Billing in GCP

Cloud Security: Introduction to security in the cloud, the shared security model, Encryption options, Authentication and authorization with Cloud IAM, Identify Best Practices for Authorization using Cloud IAM.

Cloud Network : Introduction to networking in the cloud, Defining a Virtual Private Cloud, Public and private IP address basics, Google's network architecture, Routes and firewall rules in the cloud, Multiple VPC networks, Building hybrid clouds using VPNs, interconnecting, and direct peering, Different options for load balancing.

List of Course Seminar Topics:

1. Storage Cost Optimization On Cloud
2. Cloud Security And Cryptography
3. Infrastructure As A Code (IAC)
4. Cloud Computing In Healthcare
5. Serverless Architecture
6. Deployment Of Microservices In Kubernetes Engine
7. RPA Using AWS Cloud
8. Cloud Trends In Supporting Ubiquitous Computing
9. Mobile Cloud Computing
10. Modern Data Center Architecture

List of Course Group Discussion Topics:

1. Data Storage Security in Cloud
2. Cloud Services for SMB's
3. Monitoring Services Provided by GCP and AWS
4. Docker and Kubernetes
5. SaaS vs FaaS (Function as a service)
6. Hybrid Cloud
7. GCP Vs AWS Web Service Architecture
8. Cloud based security issues and threats
9. Authentication and identity

10. Future of Cloud-Based Smart Devices

List of Home Assignments:

Design:

1. Serverless Web App to order taxi rides using AWS lambda.
2. Deploying App on Kubernetes
3. Serverless web Application (GCP Cloud Functions)
4. Demonstration of EBS, Snapshot, Volumes
5. Single Node Cluster Implementation (Hadoop)

Case Study:

1. PayU Migration to AWS
2. Cloud object storage
3. Deployment and Configuration options in AWS
4. Deployment and Configuration options in Microsoft Azure
5. Deployment and Configuration options in GCP

Blog

1. Comparing design of various cloud computing platforms
2. AWS EKS and Google Cloud Functions
3. App Engine
4. Cloud Endpoints
5. Cloud Pub/Sub

Surveys

1. Disaster Recovery in Cloud Computing
2. Cloud Economics

3. Data archiving solutions

4. Salesforce

5. Dropbox

Suggest an assessment Scheme:

MSE, ESE, GD, Seminar, HA

Text Books: ***(As per IEEE format)***

1. *Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper, "Cloud Computing for Dummies", Wiley,India.*

2. Ronald Krutz and Russell Dean Vines, "Cloud Security", Wiley-India

3. Gautam Shroff. "Enterprise Cloud Computing", Cambridge

Reference Books: ***(As per IEEE format)***

1. Barrie Sosinsky, "Cloud Computing Bible", Wiley India

2. Antohy T Velte, et.al, "Cloud Computing : A Practical Approach", McGraw Hill.

3. Michael Miller, "Cloud Computing", Que Publishing.

4. Tim Malhar, S.Kumaraswammy, S.Latif, "Cloud Security & Privacy", SPD,O'REILLY

5. Scott Granneman, "Google Apps", Pearson

Moocs Links and additional reading material:

https://nptel.ac.in/courses/106/105/106105167/

https://swayam.gov.in/nd1_noc20_cs55/preview

https://www.coursera.org/specializations/cloud-computing

https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/

https://aws.amazon.com/what-is-cloud-computing/

https://www.ibm.com/in-en/cloud/learn/cloud-computing

Course Outcomes:

1) Describe the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing

2) Explain the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc.

3) Identify problems, and explain, analyze, and evaluate various cloud computing solutions.

4) Choose the appropriate technologies, algorithms, and approaches for the related issues.

5) Display new ideas and innovations in cloud computing.

6) Collaboratively research and write a paper on the state of the art (and open problems) in cloud computing.

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO3	PO7	PO10	PO11	PSO4
3	3	2	2	1	3

CO attainment levels

CO1-1 CO2-2 CO3-3 CO4-5 CO5-4 CO6-3

Future Courses Mapping:

After completing this course different certifications courses in cloud be taken such as AWS, Azure, Google cloud certifications. One can go for higher studies in specialization of cloud

computing and allied subjects.

Job Mapping:

Cloud Architect, Cloud Engineer, Cloud Administrator, Solutions Architect - Cloud Computing - AWS / Kubernetes, Cloud Computing Technical Consultant, Associate Cloud Computing Engineer, Cloud Computing Trainer

FF No.: 654

Syllabus Template

ET4232: DEEP LEARNING

Course Prerequisites:

Linear algebra, probability theory and statistics, Digital signal processing, Computer vision

Course Objectives:

1. To present the mathematical, statistical and computational concepts for stable representations of high-dimensional data, such as images, text
2. To introduce NN and techniques to improve network performance
3. To introduce Convolutional networks
4. To introduce Sequential models of NN
5. To build deep nets with applications to solve real world problem

Credits: 2 **Teaching Scheme Theory: 2** Hours/Week

Course Relevance:

Deep learning is revolutionizing the technology and business world today. It is a subfield of machine learning concerned with algorithms to train computers to perform tasks by exposing neural networks to large amounts of data, its analysis and prediction. It is an incredibly powerful field with capacity to execute feature engineering on its own, uses multiple neural network layers to extract patterns from the data. Top applications of Deep learning involve, self-driving cars, natural language processing, robotics, finance, and healthcare.

SECTION-1

Topics and Contents

Foundations of neural networks and deep learning, Logistic regression as a neural network, different activation function, logistic regression cost function, logistic regression gradient descent, vectorizing logistic regression, forward and backward propagation, Techniques to improve neural networks: regularization and optimizations, hyperparameter tuning, batch normalization, data augmentation, deep learning frameworks, Implementation of neural network for a case study.

SECTION-1I

Convolutional Neural Networks, padding, strided convolution, pooling layers, convolutional implementation of sliding windows, Applications: object classification, object detection, face verification. ResNet, inception networks, bounding boxes, anchor boxes. Sequence modelling: recurrent nets, architecture, vanishing and exploding gradient problem, Applications & use cases.

List of Course Seminar Topics:

1. Deep learning for Stock Market Clustering
2. Application of Deep Networks in health care
3. Credit card fraud detection
4. Classification of skin cancer with deep neural networks
5. ALEXNET
6. VCGNET
7. Accelerating Deep Network Training by Reducing Internal Covariate Shift
8. Deep learning applications for predicting pharmacological properties of drugs
9. GAN (Generalised Adversial network)
10. Auto encoders

11. LSTM

List of Course Group Discussion Topics:

1. Recurrent or Recursive Networks for sequential Modelling?
2. Initializing network weights vs performance
3. Difficulty of training deep feedforward neural networks
4. Hyperparameter tuning: Is there a rule of thumb?
5. Problem of overfitting: How to handle?

6 Which cost function: Least squared error or binary cross entropy?

7. How to tackle with loss of corner information in CNN
8. Need of hundred classifiers to solve real world classification problem
9. Which optimization: Batch gradient descent of stochastic gradient descent
10. Activation functions: Comparison of trends
11. Remedy of problem of vanishing gradient and exploding gradient in RNN

List of Home Assignments:

Design:

1. Deep learning for library shelf books identification
2. Development of control system for fruit classification based on convolutional neural networks
3. Classifying movie review using deep learning
4. Sentiment analysis of the demonetization of economy 2016 India

5. Predicting Students Performance in Final Examination

Case Study:

1. Deep learning for security
2. Bag of tricks for efficient text classification
3. Convolutional Neural Networks for Visual Recognition
4. Deep Learning for Natural Language Processing
5. Scalable object detection using deep neural networks

Blog

1. Brain tumor segmentation with deep neural networks
2. Region-based convolutional networks for accurate object detection and segmentation
3. Human pose estimation via deep neural networks
4. Content Based Image Retrieval
5. Visual Perception with Deep Learning
6. Music genre classification system

Surveys:

1. Machine translation using deep learning - survey
2. Shaping future of radiology using deep learning
3. Training Recurrent Neural Networks
4. Text generation with LSTM
5. Deep learning applications in Biomedicine

Suggest an assessment Scheme:

1. Seminar – 10 Marks
2. Group Discussion – 10 Marks
3. Home Assignment – 10 Marks
4. Course Viva – 20 Marks
5. MSE – 25 Marks
6. ESE –25Marks

Text Books: (As per IEEE format)

1. Goodfellow, I., Bengio, Y., and Courville, A., Deeep Learning, MIT Press, 2016.
2. C., M., Pattern Recognition and Machine Learning, Springer, 2006.

Reference Books: (As per IEEE format)

1. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
2. Golub, G.,H., and Van Loan,C.,F., Matrix Computations, JHU Press,2013.
3. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

Moocs Links and additional reading material: www.nptelvideos.in

1. https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs11
2. https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs50

Course Outcomes: Students will be able to

1) Demonstrate understanding of a logistic regression model, structured as a shallow Neural network
2) Build and train a deep Neural Network
3) Apply techniques to improve neural network performance
4) Demonstrate understanding of functionality of all layers in a convolutional neural network
5) Implement convolutional networks for image recognition/classification tasks
6) Demonstrate Understanding of Recurrent nets and their applications

CO PO Map

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
1	4	3	3	2	4	1	-	1	1	1	-	2	1	1
2	4	3	3	2	4	2	-	1	1	1	-	2	3	3
3	4	3	3	3	4	2	-	1	1	1	-	2	3	3
4	4	3	3	3	4	2	-	1	1	1	-	2	3	3
5	4	4	3	3	4	2	-	1	1	1	-	2	3	3
6	4	4	3	3	4	2	-	1	1	1	-	2	3	3

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

CO attainment levels

Co1 -Level 3

Co2 - Level 3

Co3 - Level 5

Co4 - Level 4

Co5 - Level 5

Co6 - Level 4

Future Courses Mapping:

Advanced course on Deep learning including Autoencoders and Boltzmann machines, Reinforcement Learning etc

Job Mapping:

Deep learning engineer, Data Scientist and Algorithm Architect with industries in domains Healthcare, Industrials & Energy, Automobiles, Finance & Insurance, Human Resources, Agriculture, Cybersecurity, Ad & Marketing, Media and Entertainment, Government, Defence

FF No. : 654

Syllabus Template

CS4221::Information Retrieval

Course Prerequisites: Data Structures, Basic probability and statistics

Course Objectives:

1. To study basic principles and practical algorithms used for information retrieval
2. To introduce students about insights retrieval models, retrieval feedback and applications in web information management
3. To provide comprehensive details about various Evaluation methods.
4. To provide implementation insight about the topics covered in the course
5. To apply information retrieval in the domain of natural language processing (NLP)
6. Learn tools and techniques to do cutting-edge research in the area of information retrieval

Credits: 2 **Teaching Scheme Theory:** 2 Hours/Week

Course Relevance: This course provides an overview of the important issues in information retrieval, and how those issues affect the design and implementation of search engine software. The course emphasizes the technology used in Web search engines and the information retrieval theories and concepts that underlie all search applications. Mathematical experience including basic probability is strongly desirable.

SECTION-1

Introduction: Definition, Objectives, Search and Browse Capabilities; A Formal Document Representation, Characterization of IR Models Text operations, Pre - processing, Porter's Stemming Algorithm, term weighting techniques, Zipf's law, Heap's law

Information Retrieval Models; Boolean Retrieval, Extended Boolean Models, Vector Space Model, Probabilistic Model, Naïve Bayes, Text Classification, Document and Term Clustering, Flat and Hierarchical Clustering, Matrix Decomposition, Latent Semantic Indexing, Bayesian Model, Models for Browsing.

SECTION-II

Query Processing and Retrieval Evaluation: Digital libraries, Morphological, Lexical Analysis, Thesaurus Construction, Ontology. Retrieval Performance, Evaluation Measures for Ranked and Unranked Results Query Languages, Structural Queries, Relevance Feedback, Query Expansion

Indexing and Searching: Automatic Indexing, Inverted Files, Fast Inversion (FAST-INV) Algorithm, Signature Files, Partitioning, Tries, Suffix Trees and Suffix Arrays, PAT Tree, Distributed Indexing, Index Compression.

Web Searching: Index Construction. Search Engines, Browsing, Met searchers,

Searching using Hyperlinks, Crawling, Link Analysis, Architectures (Agents, Buses,

Wrappers/Mediators), Page Rank Algorithm, Hilltop Algorithm

List of Course Seminar Topics:

1. Parallel IR
2. Distributed IR
3. A Generic Multimedia Indexing (GEMINI)
4. Automatic image annotation and retrieval
5. Audio retrieval algorithms
6. Multimedia search engine
7. Semantic search engine
8. Text summarization
9. Cross-lingual search engine
10. Search Engine Spamming

List of Course Group Discussion Topics:

1. Querying Structured and unstructured Data
2. Relevance ranking
3. Similarity of documents metric: which one to choose
4. Measures of the effectiveness of an information retrieval system
5. Similarity-based retrieval techniques
6. Link analysis techniques
7. Crawling and near-duplicate pages
8. Personalized search
9. Collaborative filtering approaches
10. XML indexing and search

List of Home Assignments:

Design:

1. Build and evaluate a "person search engine". The search engine should automatically crawl and build textual representations of people that can be queried against. Example queries might include specific names (e.g., "george bush"), job descriptions (e.g., "car company ceos"), facts about the person (e.g., "highest paid female musician"), etc.

2.Build and evaluate a search engine that adapts to implicit user feedback. This requires developing a user interface that tracks various user behavioural signals (e.g., clicks, dwell times, mouse movement, etc.) and uses that information to improve the quality of the ranking function over time as more and more information becomes available.

3.Develop and evaluate an algorithm that will automatically summarize, each hour, the most widely discussed topics on Twitter. The summary should be short (e.g., tweet-length) and provide an adequate summary of the topic.

4.Develop and evaluate a tool that will automatically associate images with news articles. More specifically, given the text of a news article, the task is to automatically identify a single relevant that could be placed alongside the article. For example, for a news article about the positive effects of green tea, relevant images would include tea leaves, cups of tea, health related symbols/logos, etc.

5. Build and evaluate a cross-lingual search engine. For example, use Wikipedia as a source of parallel corpora.

Case Study:

1. Online multiplayer game

2. Pattern Matching techniques

3. Latent Semantic Indexing

4. Learning-based ranking algorithms

5. Classical evaluation metrics,

Blog

1. Index creation for IR system: Inverted Files

2. Index creation for IR system: Signature Files

3. Index creation for IR system: Suffix Trees and Suffix Arrays

4.Development of semantic search engine to deal with polysemy, synonymy issues of text documents

5. Development of text search engine

Surveys

1. Crawling and near-duplicate pages

2.Content- based filtering

3. Unified filtering

4.Topic detection and tracking

5. Cross language information retrieval

Suggest an assessment Scheme:

MSE,ESE, GD, Seminar, HA

Text Books: ***(As per IEEE format)***

1. *Ricardo Baeza-Yates, Berthier Ribeiro-Neto, "Modern Information Retrieval The Concepts and Technology behind Search", , Pearson Education: New Delhi, 2007*

2. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze "Introduction to Information Retrieval", Cambridge University Press, 2012

Reference Books: ***(As per IEEE format)***

1. Gerald Kowalski, Mark Maybury, "Information Storage and Retrieval Systems Theory and Implementation", Springer Pvt. Ltd., 2006

2. William Frakes, Ricardo Baeza-Yates, "Information Retrieval Data Structures & Algorithms" ,Pearson Education, 2008

3. C. J. Van Rijsbergen, "Information Retrieval", Information Retrieval Group, University of Glasgow

Moocs Links and additional reading material:

https://nptel.ac.in/courses/106/101/106101007/

http://www.dcs.gla.ac.uk/Keith/Preface.html

https://nlp.stanford.edu/IR-book/pdf/01bool.pdf

https://www.coursera.org/learn/google-cbrs-cpi-training

https://www.coursera.org/learn/text-retrieval

Course Outcomes:

Students will be able to

1. Describe various information retrieval system architectures and models

2. Validate retrieval performance of an information retrieval system

3.Construct various indexes using suitable techniques

4. Apply sequential search and pattern matching techniques

5. Illustrate working of parallel, distributed and multimedia information retrieval system

6. Explain various information retrieval algorithms and different types of queries

CO PO Map

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO5	PO7	PO10	PO11	PSO4
3	3	2	2	1	3

CO attainment levels

CO1- 3 CO2- 4 CO3- 2 CO4- 5 CO5-1 CO6-3

Future Courses Mapping:

Applied Natural Language Processing

Job Mapping:

Applied Scientist

Data Engineering

Syllabus Template FF654

CS4209::Parallel Computing

Course Prerequisites: Computer Organization, Operating System, Design & Analysis of Algorithms, Data Structure

Course Objectives: Students will be able to

1. To introduce the basic concepts of parallel computing
2. To understand various GPU Architecture.
3. To write CUDA programs for parallel implementation
4. To organize the memory management in GPU
5. To optimize parallel programs on GPU using CUDA
6. To solve the scientific problems using GPUs

Credits: 2 **Teaching Scheme** **Theory: 2** Hours/Week

Course Relevance: Parallel computing, on the other hand, uses multiple processing elements simultaneously to solve a problem. This is accomplished by breaking the problem into independent parts so that each processing element can execute its part of the algorithm simultaneously with the others. This course is required in the industry & used to set up data centres.

SECTION-1

Introduction to Parallel Computing: Motivating Parallelism, Scope of Parallel Computing, Parallelism vs Concurrency, Types and levels of parallelism, Flynn's classification, Amdhal's law; Parallel computer architectures : PRAM, Distributed memory systems ,Shared memory systems and cache coherence, Concept of thread and process, programming parallel computers, Parallel computing architectures, interconnection networks, Modern GPU architecture (in brief), Performance comparison: Speedup, Gain time and scalability.

Parallel Programming Model: Common Unified Device Architecture (CUDA), CUDA programming model, Concept of grid, block and thread, thread index generation, warp, kernel & kernel launch. Programming for GPU's in C/C++ using CUDA API: Memory transfers, Writing and executing kernel functions, Writing device functions, Thread synchronization, Data Dependences and Race Conditions, Organizing Parallel Threads.

SECTION-1I

Topics and Contents

GPU Architecture: GPU architecture, Overview of the graphics pipeline, Components of GPU: Parallel streaming processors, Multiprocessors, Shared instruction caches ,Memory hierarchy – Global, Constant, Shared, and Texture memory; Case studies: NVIDIA Kepler K20/K40/K80/GP100/GV100/ Ampere.

Memory Organization and Optimization: Global, Shared, constant and texture memory. Memory coalescing, memory banks and bank conflicts, Page locked host memory. Reduction operation, CUDA code optimization. Need of profilers and analyzers, Introduction to CUDA Tools: MemCheck, Command line & Visual Profilers.

Problem solving using GPUs: Single vs. double precision, light weight scientific computing exercises, Image processing applications, Matrices etc .

List of Home Assignments:

Design:

1. Parallelizing Search Trees for Chess
2. Parallel Algorithm for Searching
3. Parallel Algorithm for sorting
4. Parallel Algorithm for Data mining
5. Parallel Algorithm for Image Processing

Case Study:

1. Nvidia DGX2
2. Jetson nano Developer Kit
3. GPU Accelerated Apache Spark
4. The Jetson Xavier NX Developer Kit
5. NVIDIA Ampere architecture

Blog

1. Cuda library
2. Turing mesh shaders
3. Low level GPU Virtual memory management
4. Memory Hierarchy of GPU
5. Comparison of Various GPUs

Surveys

1. Smart Hospitals through AI with GPUs
2. Clara Models to help fight with COVID 19
3. GPU Accelerated Molecular Dynamics Applications
4. Medical Imaging applications of GPU
5. Ray Tracing Applications of GPU

Suggest an assessment Scheme:

MSE(30)+ESE(30)+HA(10)+CVV(20)

Text Books: ***(As per IEEE format)***

1. Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar; Introduction to parallel computing; second edition., Addison- Wesley, 2003, ISBN: 0201648652

2*David Kirk, Wen-mei HwuCUDA: Programming Massively Parallel Processors: A Hands-On Approach. © ELSEVIER Inc.*

3*Jason Sanders and Edward KandrotCUDA by Example: An Introduction to General-Purpose GPU Programming"*

Reference Books: ***(As per IEEE format)***

1. *Hwang and Briggs,"Computer Architecture and Parallel Processing", Tata McGraw Hill Publication ISBN 13: 9780070315563.*

2.John Cheng, Max Grossman, Ty McKercherProfessional CUDA C Programming,

3. *CUDA C PROGRAMMING GUIDE by NVIDIA*

Moocs Links and additional reading material:www.nptelvideos.in

http://developer.nvidia.com/

Course Outcomes:

The student will be able to –

1)Recognize various parallel computing architectures and their fundamentals
2)Investigate parallel solutions to complex real world problems
3)Code the parallel programs on GPU using CUDA
4)Evaluate the performance on various GPU architectures
5)Optimize the parallel programs on GPU using CUDA
6)Design and develop new solutions to research problems

CO PO Map
CO1 –PO3(3)

CO2 –PO5(3)
CO3 –PO7(2)
CO4 –PO11(1)
CO5-PO12(1)
CO6-PSO3(3)

CO attainment levels
CO1 –3
CO2 -3
CO3 –2
CO4 –1
CO5-1
CO6-3

Future Courses Mapping:

High Performance Computing
Distributed Computing

Job Mapping:
What are the Job opportunities that one can get after learning this course
Full Stack Architect-GPU
Developer Technology Engineer
Software Engineer Cloud
Data Analytics Engineer
Cloud Developer
Senior Software Engineer
HPC GPU Application Developer & Consultant
GPU Programming Professional
GPU Performance Analysis Lead / Architect
GPU Advocate Associate

CS4225/CS4226::MAJOR PROJECT

Course Prerequisites: Project Based Learning

Credits:.10..... Teaching Scheme Theory: …20... Hours/Week

Aim

This course addresses the issues associated with the successful management of a project. The course emphasizes project life cycle phases requirement engineering, system analysis and system design. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and working with peers to create desired outcomes. The Project Work can lead to:

- Transform existing Ideas into conceptual models.
- Transform conceptual models into determinable models.
- Use determinable models to obtain system specifications.
- Select optimum specifications and create physical models.
- Apply the results from physical models to create real target systems.

Project Group and Topic Selection and Synopsis:

The project work needs to be undertaken by a group of maximum FOUR and minimum of THREE students. The Project work will be jointly performed by the project team members. The student needs to identify a technological problem in the area of Computer Engineering or Information Technology of their choice and address the problem by formulating a solution for the identified problem. The Project Group will prepare a synopsis of the project work which will be approved by the concerned faculty member. The project should not be a reengineering or reverse engineering project. In some cases, reverse engineering projects will be permissible based on the research component involved in it. The project work aims at solving a real world technical problem. Hence ample literature survey is required to be done by the students. Application-oriented projects will not be acceptable. Low-level custom User Interface development and its allied mapping with a particular technology will not be accepted.

Overview of the Course:

1. The Student Project Group is expected to make a survey of situation for identifying the requirements of selected Technological Problem. The Student Project Group will be monitored by Internal Guides and External Guides (if any).
2. The project requires the students to conceive, design, implement and operate a mechanism (the design problem). The mechanism may be entirely of the student's own design, or it may incorporate off-the-shelf parts. If the mechanism incorporates off-the-shelf parts, the students must perform appropriate analysis to show that the parts are suitable for their intended purpose in the mechanism.

3. The project must be based on a Fresh Idea or Implementation of a Theoretical Problem – meaning that there is not a known Solution to the design problem Or Create a Better Solution.
4. The project must have an experimental component. Students must conceive, design, implement and operate an appropriate experiment as part of the project. The experiment might be to collect data about some aspect of the design (i.e., to verify that the design will work as expected). Alternatively, the experiment could be to verify that the final mechanism performs as expected.
5. Upon receiving the approval, the Student Project Group will prepare a preliminary project report consisting , Feasibility Study Document, System Requirement Specification, System Analysis Document, Preliminary System Design Document. All the documents indicated will have a prescribed format.

6. Upon project completion, the Student Project Group will prepare a detailed Project Report consisting Semester I Preliminary Project document along with Detailed System Design Document, Implementation and Testing Document with conclusion and future scope of the Project Work. All the documents indicated will have a prescribed format. The Project Report ideally should consist of following documents : (Exceptions may be there based on the nature of the project, especially if some of the following documents are not applicable to a particular project as determined by the project guide, coordinator and head of department).

Sr.	Project Item
1	Project Cover Front Page
2	Project Completion Certificate [Institute]
3	Project Completion Letter [In case of Sponsored Projects]
4	Acknowledgments
5	Table of Contents
6	List of Figures
7	List of Tables
8	Project Synopsis [Problem Background, Existing System Details, Proposed Solution]
9	Feasibility Study Report
10	Project Plan
11	System Requirement Specification
12	System Analysis Document: UML Use Case Diagrams
13	System Analysis Document: UML Sequence Diagrams
14	System Analysis Document: UML State Diagrams

15	System Design Document with Module Specifications
16	System Implementation
17	System Testing and Experimental Findings
18	Conclusion
19	References

7. The Project Work will be assessed jointly by a panel of examiners consisting faculty and industry experts. The Project Groups will deliver the presentation and demonstration of the Project Work which will be assessed by the panel.
8. The Student Project Group needs to actively participate in the presentation. The panel of examiners will evaluate the candidate's performance based on presentation skills, questions based on the Project Work and overall development effort taken by the candidates.

Note:
The student needs to design and develop solution for the identified technological problem in the area of Computer Engineering or Information Technology of their choice. The Project Implementation needs to be completed using best possible use of available technologies as applicable to deal with the complexity of the project. The Project Group will prepare a detailed report of the project work which will be approved by the concerned faculty member. The Project Report need to be submitted both in Hard form and Soft form in CD. The Soft Copy of the Project Report must accompany other project deliverables as well.

Assessment: MSE and ESE
1. Mid Semester Assessment – 50 Marks to be converted to 30 Marks.
2. End Semester Assessment – 100 Marks to be converted to 70 Marks.

Mid Semester Assessment

Sr. No.	Parameter	Marks
1	Problem Statement	10
2	Literature Review	10
3	Group formation and identification of individual responsibility	10
4	Objective of Project activity	10
5	Knowledge of domain, latest technology and modern tools used /to be used	10
	TOTAL	**50**

End Semester Assessment

Sr. No.	Parameter	Marks

1	Realization of project as per problem statement	10
2	Design, Testing / Experimentation, Analysis / Validation	30
3	Documentation and Report Writing	20
4	Quality of Work	15
5	Performance in Question & Answers Session	15
6	Regular interaction with guide	10
	TOTAL	**100**

Course Outcomes:

Upon completion of the course, graduates will be able to -

CO1: Identify the real life problem from societal need point of view

CO2: Prepare the requirement engineering , feasibility analysis documents

CO3:Form the teams and share responsibilities according to individual skill strengths

CO4:Create design documents to build software solutions

CO5:Develop software solutions based on standard engineering specifications

CO6: Perform the verification and validation up to the mark

CO attainment levels

CO1 -4 CO2 –2 CO3-4 CO4-5 CO5 -1 CO6-3

CO-PO Map:

.

CO1	CO2	CO3	CO4	CO5	CO6
PO2	PO4, PO5, PO12	PO6, PO9	PO3	PO5, PO8	PSO3, PSO4
3	3, 3, 2	2, 2	2	2, 2	3, 3