Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Structure & Syllabus

B.Tech. Computer Science & Engineering (Artificial Intelligence)

With Effect from Academic Year 2024-25

Prepared by: - Board of Studies in CSE (Artificial Intelligence)

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

Chairman – BOS **Chairman – Academic Board**

Vision of the Institution

"To be a 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 Citizens.

Vision of the Department

“To be a center of academic excellence in Artificial Intelligence and develop AI innovators”.

Mission of the Department

•To ensure and enhance students' employability and entrepreneurial skills through knowledge of principles of Computing, Artificial Intelligence and soft skills.

•To maintain a high standard of academic excellence by offering a rigorous and state-of-the-art curriculum and active participation of industry.

•To cultivate research culture by investigating complex problems by using principles of Computing and Artificial Intelligence knowledge.

•To instill a sense of responsibility and ethics to make students’ as responsible citizens.

List of Program Outcomes [PO]

PO	PO Statements
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.

Program Specific Objectives (PSO)

PSO1: Demonstrate proficiency in essential concepts of computer science and programming solutions.

PSO2: Formulate robust software design, execution, and testing strategies employing a software paradigms and Artificial Intelligence knowledge to solve real word problems.

PSO3: Adapt and exhibit expertise in evolving areas of computer science, engineering and technology.

Program Educational Objectives (PEOs)

PEO	PEO Focus	PEO Statements- AI
PEO1	**Core competence**	Demonstrate core **competence** in principles of computing and AI based technologies'.
PEO2	**Breadth**	Apply AI principles, methodologies, algorithms, and tools, to effectively **design**, develop, and implement AI-driven solutions.
PEO3	**Professionalism**	Excel as professionalism with the necessary **soft skills** to work collaboratively in interdisciplinary teams.
PEO4	**Learning Environment**	Aim for **continuing education** and **entrepreneurship** in emerging areas of computing and Artificial Intelligence.

Course Name Nomenclature as per NEP (For FY and SY)

BSC: Basic Science Course	MDOE: Multi Disciplinary Open Elective
ESC: Engineering Science Course	CC: Co-curricular Course
PCC: Program Core Course	HSSM: Humanities Social Science and Management
PEC: Program Elective Course	IKS: Indian Knowledge System
ELC: Experiential Learning Course	FP: Field Project
MD: Multi Disciplinary	INT: Internship

Nomenclature for Teaching and Examination Assessment Scheme AY 2024-25

Sr No.	Category	Head of Teaching/ Assessment	Abbreviation used
1	Teaching	Theory	Th
2	Teaching	Laboratory	Lab
3	Teaching	Tutorial	Tut
4	Teaching	Open Elective	OE
5	Teaching	Multi Disciplinary	MD
6	Teaching	Computer Science and Engineering – Artificial Intelligence	CI
7	Assessment	Laboratory Continuous Assessment	CA
8	Assessment	Mid Semester Assessment	MSA
9	Assessment	End Semester Assessment	ESE
10	Assessment	Home Assignment	HA
11	Assessment	Course Project	CP
12	Assessment	Group Discussion	GD
13	Assessment	PowerPoint Presentation	PPT
14	Assessment	Class Test –1	CT1
15	Assessment	Class Test –2	CT2
16	Assessment	Mid Semester Examination	MSE
17	Assessment	End Semester Examination	ESE
18	Assessment	Written Examination	WRT
19	Assessment	Multiple Choice Questions	MCQ
20	Assessment	Laboratory	LAB

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

Branch: Computer Science & Engineering (Artificial Intelligence) **Year:** S.Y. **A.Y.:** 2024-25 **Module:** III

Sub No.	Sub Code	Subject Name	Teaching Scheme (Hrs/Week)				Examination Scheme										Credits
			Th	Lab	Tut	CA	Test-1	MSA	Test-2	ESA						Total	
						LAB (%)	CT1 (%)	MSE (%)	CT2 (%)	HA (%)	LAB (%)	CP (%)	PPT /GD (%)	CVV (%)	ESE (%)		
S1	MD2201	ESC: Data Science	2	2	1	10	-	-	-	-	-	20	20 (PPT)	20	30 (WRT)	100	4
S2	CS2221	MD: Internet of Things	2	2	1	10	-	-	-	-	-	20	20 (GD)	20	30 (MCQ)	100	4
S3	CS2218	PCC: Object Oriented Programming	2	2	1	10	-	-	-	-	50	20	-	20	-	100	4
S4	CS2227	PCC: Database Management System	2	2	1	10	-	-	-	-		20	20 (PPT)	20	30 (WRT)	100	4
S5	CS2249	BSC: Mathematics for AI	2	0	0	-	35 (WRT)	-	35 (WRT)	-	-	-	-	30	-	100	2
S6	CI2006	CC: Design Thinking 3	-	-	-	-	-	-	-	-	-	-	-	-	100	100	1
S7	CI2007	PCC: Engineering Design and Innovation-III	-	8	-	-	-	30	-	--	-	-	-	-	70	100	4
S8	SH2001	BSC: Reasoning and Aptitude Development	-	-	-	-	-	-	-	-	-	-	-		-	-	1
		Total	**10**	**16**	**4**	**40**	**35**	**30**	**35**	**00**	**50**	**80**	**60**	**110**	**260**	**700**	**24**

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

Branch: Computer Science & Engineering (Artificial Intelligence) **Year:** S.Y. **A.Y.:** 2024-25 **Module:** IV

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme										Total	Credits
			Th	Lab	Tut	CA	Test-1	MSA	Test-2	ESA						Total	
						LAB (%)	CT1 (%)	MSE (%)	CT2 (%)	HA (%)	LAB(%)	CP (%)	PPT /GD (%)	CVV (%)	ESE (%)		
S1	CI2001	PCC: Data Structures	2	2	1	10	-	-	-	-	50	20	-	20	-	100	4
S2	CI2002	PCC: Artificial Intelligence	2	2	1	10	-	-	-	20		20		20	30 (WRT)	100	4
S3	CI2003	PCC: Operating Systems	2	2	1	10	-	-	-	-	50	20		20		100	4
S4	CI2004	PEC: Data Visualization	2	2	1	10	-	-	-	-	-	20	20 (PPT)	20	30 (WRT)	100	4
S5	CI2005	ESC: Theory of Computation	2	-	-	-	35 (WRT)	-	35 (WRT)-	-	-	-	-	30	-	100	2
S6	CI2006	CC: Design Thinking3	-	-	-	-	-	-	-	-	-	-	-	-	100	100	1
S7	CI2007	PCC: Engineering Design and Innovation-III	-	8	-	-	-	30	-	-	-	-	-	-	70	100	4
S8	SH2001	BSC: Reasoning and Aptitude Development	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1
		Total	**10**	**16**	**4**	**40**	**35**	**30**	**35**	**20**	**100**	**80**	**20**	**110**	**230**	**700**	**24**

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

Branch: Computer Science & Engineering (Artificial Intelligence) **Year:** T.Y. **A.Y.:** 2024-25 **Module: V**

			Teaching Scheme (Hrs/Week)			Examination Scheme									Credits
Sub No.	**Sub Code**	**Subject Name**	**Th**	**Lab**	**Tut**	**CA**	**MSA**	**ESA**						**Tota l**	
						LAB (%)	**MSE (%)**	**HA (%)**	**LAB (%)**	**CP (%)**	**PPT /GD (%)**	**CVV (%)**	**ESE (%)**		
S1	CI3001	Computer Network Technology	2	2	1	10	-	-	50	20	-	20	-	100	4
S2	CI3002	Design and Analysis of Algorithms	2	2	1	10	-	20	-	20	-	20	30 (WRT)	100	4
S3	CI3003	Artificial Neural Networks	2	2	1	10	-	-	-	20	20 (PPT)	20	30 (WRT)	100	4
S4	CI3004	Cloud Computing	2	2	1	10	-	-	-	20	20 (PPT)	20	30 (MCQ)	100	4
S5	CI3005	Design Thinking – 5	-	-	-	-	-	-	-	-	-	-	100	100	1
S6	CI3006	Engineering Design and Innovation – 3	-	12	-	-	30	-	-	-	-	-	70	100	6
S8	SH3001	Reasoning and Aptitude Development	-	-	-	-	-	-	-	-	-	-	-	-	1
S9	AC*	Audit Courses*	-	-	-	-	-	-	-	-	-	-	-	-	0
		Total	**8**	**20**	**4**	**40**	**30**	**20**	**50**	**80**	**40**	**80**	**260**	**600**	**24**

Audit Courses for Third Year: Module -V and Module-VI:
1. MD3144 Basics of Game Development--offered by Zensar Technologies,
2 MD3146: Main Frame Technologies –offered by BMC

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

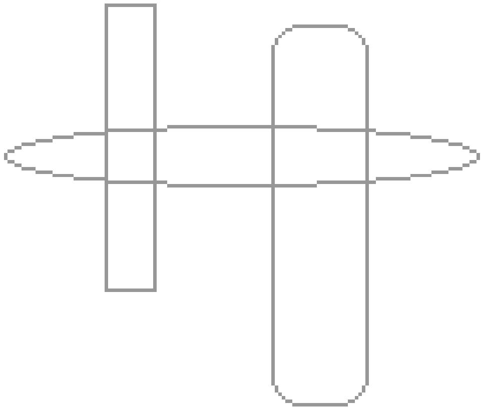
Branch: Computer Science & Engineering (Artificial Intelligence) **Year:** T.Y. **A.Y.:** 2024-25 **Module: VI**

Sub No.	Sub Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme									Credits
			Th	Lab	Tut	CA	MSA	ESA						Total	
						LAB (%)	MSE (%)	HA (%)	LAB (%)	CP (%)	PPT /GD (%)	CVV (%)	ESE (%)		
S1	CI3007	Software Engineering	2	2	1	10	-	-	-	20	20 (GD)	20	30 (MCQ)	100	4
S2	CI3008	Cyber Security and Blockchain	2	2	1	10	-	20	-	20		20	30 (WRT)	100	4
S3	CI3009	Deep Learning	2	2	1	10	-	-	50	20	-	20	-	100	4
S4	Coursera	Coursera Courses*	-	-	-	-	-	-	-	-	-	-	100	100	4
S5	CI3010	Design Thinking – 5	-	-	-	-	-	-	-	-	-	-	100	100	1
S6	CI3011	Engineering Design and Innovation – 3	-	12	-	-	30	-	-	-	-	-	70	100	6
S8	SH3001	Reasoning and Aptitude Development	-	-	-	-	-	-	-	-	-	-	-	-	1
		Total	**6**	**18**	**3**	**30**	**30**	**20**	**50**	**60**	**20**	**60**	**330**	**600**	**24**

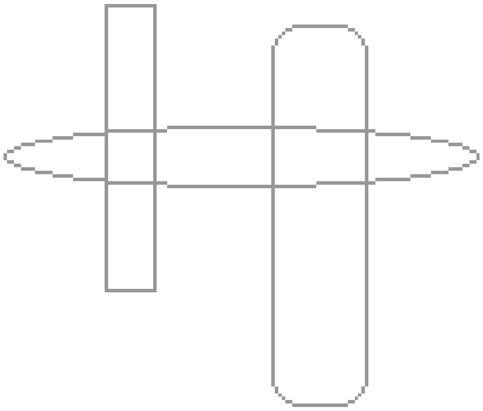
*Coursera - Subject Code	Coursera Subject Name	Coursera Subject Code	Coursers Subject Name
MD4228	IBM Full Stack Software Developer	MD4245	IBM Data Science
MD4230	IBM Back-End Developer	MD4247	IBM Data Warehouse Engineer

MD4248	IBM DevOps and Software Engineering	MD4257	IBM Mainframe Developer
MD4262	Salesforce Sales Development Representative	MD4269	Google UX Design
MD4238	Microsoft Cybersecurity Analyst		Cadence Specialization
MD4243	IBM Data Engineering		

1.



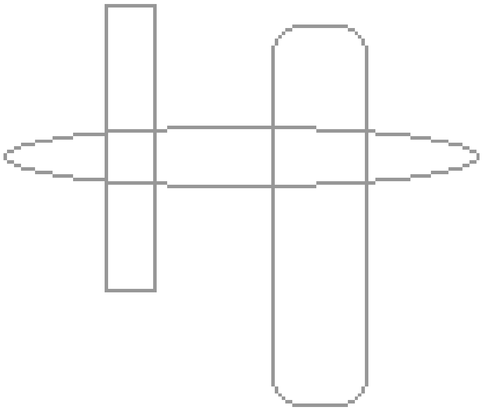
INDEX



S. Y. B. Tech. Computer Science & Engineering (Artificial Intelligence)

AY 2024-25

Module III Course Content



Syllabus Template **FF No. : 654**

MD2201: Data Science

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**
Tut: 1 Hour/Week
Lab: 2 Hours/Week

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

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: Topics/Contents

Data science definition, raw data, processed data and their attributes, meta data, data cleaning, data science pipeline. **(3 Hours)**

Normal distribution, evaluating normal distribution, Binomial distribution, confidence Intervals, central limit Theorem, hypothesis testing **(6 Hours)**

Vector norms, Unconstrained Optimization **(4 Hours)**

Section 2: Topics/Contents

Simple and multiple linear regressions; Logistic regression, non-linear regression, polynomial regression **(4 Hours)**

Nearest Neighbor Classification – Knn approach, branch and bound algorithm, projection algorithm; Naïve Bayes Classification; Classification using decision trees, divisive and agglomerative clustering, K-means clustering **(6 Hours)**

Evaluation of model performance – Confusion matrices, sensitivity, specificity, precision, recall, F-measure, Classifier performance measurement metrics – Training & Testing strategies, Resubstitution, Hold-out, Cross validation, Bootstrap **(3 Hours)**

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 Practical's: (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: 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.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Presentation: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (Written): 60 Marks converted to 30 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 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 and data visualization techniques
2. Perform descriptive and inferential statistical analysis
3. Utilize appropriate distance metrics and optimization techniques

4. Implement supervised algorithms for classification and prediction
5. Implement unsupervised classification algorithms
6. Evaluate the performance metrics of supervised and unsupervised algorithms

CO-PO Mapping:

	Program Outcomes (PO)												PSO			
CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	2	2	1			2										
CO2	3	3	3	3	2							2	2			
CO3	3	3	3	3	2							2	2			
CO4	3	3	3	2	2								3		2	
CO5	3	3	3	2	3								3		2	
CO6	2	2	2													
Average	**2.66**	**2.66**	**2.5**	**2.5**	**2.25**	**2.0**						**2.0**	**2.5**		**2.0**	

Future Courses Mapping:

Deep Learning, Reinforcement Learning, Big Data, Data Mining, Information Retrieval, Recommendation Systems, Cloud Computing – AWS, IOT

Job Mapping:

Job opportunities that one can get after learning this course

Data Scientist, Data Analyst, AI Engineer, Data Architect, Data Engineer, Statistician.
Database Administrator, Business Analyst, Business Intelligence Developer, Infrastructure Architect
Enterprise Architect, Machine Learning Engineering, Machine Learning Scientist

Syllabus Template **FF No.: 654**

MD2221: Internet of Things

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**
Tut: 1 Hour/Week
Lab: 2 Hours/Week

Course Prerequisites:

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

Course Relevance:

The Internet of Things is transforming our physical world into a complex and dynamic system of connected devices on an unprecedented scale. Interne 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: Topics/Contents

Introduction to IoT

Physical Design of IOT, Logical Design of IOT, IOT Enabling Technologies, IOT Levels &

Deployment Templates **(4 Hours)**

IOT Platform Design Methodology

IoT Design Methodology Steps, Home Automation Case Study, Smart Cities, Health Care, Agriculture **(4 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. **(6 Hours)**

Section 2: Topics/Contents

Introduction to Wireless Sensor Network

Sensor Node, Smart Sensor Network, Wireless Sensor Network, RFID - Principles and Components, Node MCU **(4 Hours)**

Connectivity Technologies

Network Configuration in IoT, IoT Stack and Web Stack, IEEE 802.15.4 Standard, Zigbee, Bluetooth, MQTT, Cloud Architecture and Types, Cloud Service Providers **(8 Hours)**

Case Studies (Any Two from following List to be covered)

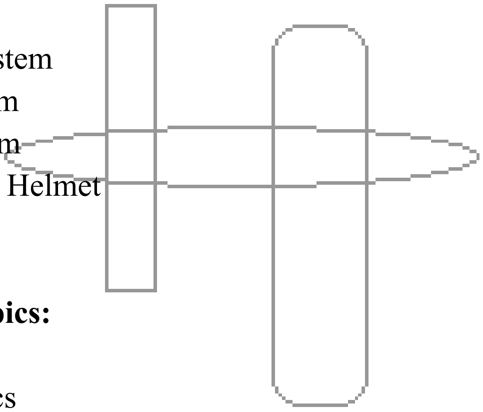
Smart lighting, Home Intrusion Detection, Smart Parking, Weather Monitoring System, Weather Report Bot, Air Pollution Monitoring, Forest fire Detection, Smart Irrigation, IoT Printer, IoT in Manufacturing Industry, IoT in Process Industry, IoT in Quality, Control Applications in Industry, IoT in Material Handling System in Industry, IoT in Automobile Industry, Navigation System, Connected Vehicles, Industry 4.0 **(4 Hours)**

List of Practical’s: (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. 5GinIoT
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

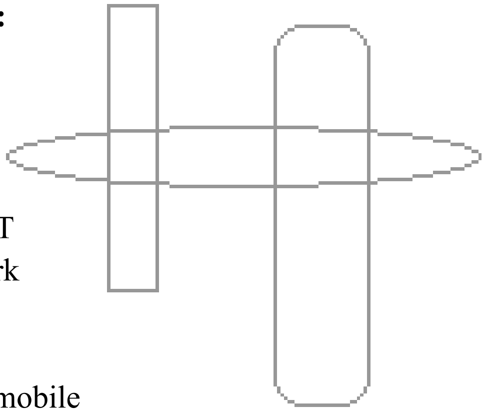
List of Group Discussion Topics:

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 Industry4.0

Surveys:

1. Autonomous Vehicles
2. ListofIndiancompanieswhichofferIoTsolutionsforagricultureandfarming.Describetheproble m 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 ,justout side(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.

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.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Group Discussion: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (MCQ): 60 Marks converted to 30 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

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

1. *Arshdeep Bahga and Vijay Madisetti, Internet of Things: A Hands-on Approach", (Universities Press)*
2. *PethuruRajandAnupamaC.Raman,"TheInternetofThings:EnablingTechnologies,Platforms,and Use Cases", (CRC Press)*

Reference Books:

1. *Adrian McEwen, Hakim Cassimally "Designing the Internet of Things", Wiley*
2. *OvidiuVermesan&PeterFriess"InternetofThingsApplications-FromResearchandInnovationtoMarket Deployment", ISBN:987-87-93102-94-1,RiverPublishers*
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 Attainmentlevel:2)
2. Recognize IoT Design Methodology Steps(COAttainmentlevel:3)
3. Select sensors for different IoT applications (COAttainmentlevel:3)
4. Analyze fundamentals of networking (COAttainmentlevel:4)
5. Apply basic Protocols in IoT (CO Attainmentlevel:4)
6. Provide IoT solutions practically with the help of case study(COAttainmentlevel:5)

CO-PO Mapping [Department]

	Program Outcomes (PO)												PSO			
CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	2											2	2			
CO2	2		3	2			2				1	2	2	3	3	3
CO3	2	2	3								1	2	2	3		
CO4	2	3										2	2	3		
CO5	2	2	3		3		2					2	2	3	3	3
CO6	2	3	3	2	3	2	2	2		2		2	2	3	3	3
Average	**2.0**	**2.5**	**3.0**	**2.0**	**3.0**	**2.0**	**2.0**	**2.0**		**2.0**	**1.0**	**2.0**	**2.0**	**3.0**	**3.0**	**3.0**

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, IoT Software Developer, IoT Solution Architect, IoT Service Manager and many more.

Syllabus Template

FF No.: 654

CS2218: Object Oriented Programming

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**
Tut: 1 Hour/Week
Lab: 2 Hours/Week

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

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: Topic and Contents

Introduction:

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

Java overview: Classes and Objects, Java object storage, Access Modifiers, this reference, main method, Static vs Instance block, Static methods vs Instance methods in Java.

Constructors: Constructors in Java, Default constructor, Parameterized constructor.

Input and Output: Byte Stream vs Character Stream, use of Scanner Class.

Arrays in Java: Arrays in Java, initialization, Default Array values, multi-dimensional array, java.util .Arrays class, string class, string buffer, string builder.

Methods in Java: Methods, Parameters passing, Returning Multiple values.

Inheritance: Inheritance in Java, Types, Constructor in Inheritance, Using final with Inheritance, Accessing superclass member, 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

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

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

Collection in Java: Collections Class, Using Iterators, Iterator vs Foreach, Array List, Vector, Map, Set.

Multithreading: Thread life Cycle, Thread Priority, Thread Methods.

File Handling: File Processing, Primitive Data Processing, Object Data Processing, Connecting Java with database (JDBC/ODBC).

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

File Handling: File Processing, Primitive Data Processing, Object Data Processing, Connecting Java with database (JDBC/ODBC).

Section 2: Topic and Contents

Introduction:

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

Java overview: Classes and Objects, Java object storage, Access Modifiers, this reference, main method, Static vs Instance block, Static methods vs Instance methods in Java.

Constructors: Constructors in Java, Default constructor, Parameterized constructor.

Input and Output: Byte Stream vs Character Stream, use of Scanner Class.

Arrays in Java: Arrays in Java, initialization, Default Array values, multi-dimensional array, java.util .Arrays class, string class, string buffer, string builder.

Methods in Java: Methods, Parameters passing, Returning Multiple values.

Inheritance: Inheritance in Java, Types, Constructor in Inheritance, Using final with Inheritance, Accessing superclass member, 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

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

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

Collection in Java: Collections Class, Using Iterators, Iterator vs Foreach, Array List, Vector, Map, Set.

Multithreading: Thread life Cycle, Thread Priority, Thread Methods.

File Handling: File Processing, Primitive Data Processing, Object Data Processing, Connecting Java with database (JDBC/ODBC).

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

List of Practical's:

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 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 comparing 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 Buffer Reader Class
10. String, String buffer and String builder

List of Group Discussion 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 comparing 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 Buffer Reader Class
10. String, String buffer and String builder

List of Course Projects:

1. Airline reservation system
2. Course management system
3. Data visualization software
4. Electricity billing system
5. e-Healthcare management system
6. Email client software
7. Library management system
8. Network packet sniffer
9. Online bank management system
10. Online medical management system

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 aArrayList 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: 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.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Laboratory Practical: End Semester Examination: 100 Marks converted to 50 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

Text Books:

1. 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

CO-PO Mapping:

	Program Outcomes (PO)												PSO			
CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	3	2	2									2	3			
CO2	3	3	3	1	1	2	2	2			2	2	3		3	
CO3	3	2	3	1	1		2					2	3		3	3
CO4	3	3	3	1							2	2	3		3	
CO5	3	3	3	1	1						2	2	3		3	
CO6	3	2	3									2	3		3	
Average	**3**	**2.5**	**2.83**	**1.0**	**1.0**	**2.0**	**2.0**	**2.25**			**2.0**	**2.0**	**3.0**	**3.0**	**3.0**	**3.0**

Future Courses Mapping:

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

Job Mapping:

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

Syllabus Template **FF No. : 654**

CS2227: Database Management Systems

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**
Tut: 1 Hour/Week
Lab: 2 Hours/Week

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.

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 1: Topics/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 2: Topics/Contents

Storage and Querying: Storage and File structures, Indexed Files, Single Level and Multi Level Indexes; Query Processing, Query Optimization, Parquet file format.
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, Introduction to Elastic Search index.
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

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.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Presentation: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (Written): 60 Marks converted to 30 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

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 Mapping - Department

	Program Outcomes (PO)												PSO			
CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
CO1	2	3	3		2		2			2	2	2	3		2	
CO2	2	2	3		2		2			2	2	2	3	3	2	3
CO3	2	3	3				2			2	2	2	3	3	2	
CO4	2	3	3	2						2		2	3			
CO5	2	-	3									2	3			
CO6	2	-	3									2	3			
Average	**2.0**	**2.75**	**3.0**	**2.0**	**2.0**		**2.0**			**2.0**	**2.0**	**2.0**	**3.0**	**3.0**	**2.0**	**3.0**

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

Syllabus Template **FF No.: 654**

CS2249: Mathematics for AI

Credits: 2 **Teaching Scheme Theory: 2 Hours/Week**

Course Prerequisites: Engineering Mathematics

Course Objectives: Students will learn

1. Prepositional logic, predicate logic to reason about the relationships between propositions and predicates
2. Nonmonotonic reasoning to capture and represent defeasible inferences
3. Different trees and search algorithms to find the best possible solution inferences
4. Bayesian networks to solve a problem which has uncertainty due to multiple events
5. Fuzzy logic use to imitate human reasoning and cognition
6. Finding minima to aids in the discovery of a function's lowest extreme values for optimization

Course Relevance:

Since some of the mathematics used in AI is not part of a standard undergraduate curriculum, you'll be learning mathematics and seeing how it's used in AI at the same time.

Section 1: Topics/Contents

Unit-I Logic and AI **04 Hours**

What Is Mathematical Logic? Logic and AI, Propositional Logic, Syntax and Semantics of Propositional Logic, Predicate Logic, Syntax and Informal Semantic of Predicate Logic, The Theory of Resolution. Truth versus Proof, Truth, Proof, Resolution and Propositional Calculus, The Resolution Method, Resolution of Horn Clauses, First-Order Predicate Calculus, Skolemization, Unification, Resolution, Soundness and Completeness, Decidability

Unit-II Nonmonotonic Reasoning **05 Hours**

Types of Qualitative Nonmonotonic Reasoning, How Well Do Nonmonotonic Methods Work? Default Reasoning, Normal Default Theories, Other Modifications of Logic, Circumscription, Modal and Autoepistemic Logics, Rule Systems, Basic Concepts of Monotonic Systems, Forward versus Backward Chaining, Negation, Limiting the Effects of Contradictions, Nonmonotonicity, Semantic Nets, Frames, Manipulating Simple Inheritance Systems, Defeasible Reasoning, The Syntax of Defeasible Reasoning, The Laws of Defeasible Reasoning

Unit-III Tree and Search **05 Hours**

Problem Spaces and Search Trees, Decision Trees, Simple Search, Iterative-Deepening Search, Heuristic Search - Partial Search: Use a Limited-Depth Search Make a Decision and Start Again, Limited-Depth Search, AND/OR Trees, Alpha-Beta Pruning, Search in game of Chess and Maze

Section2: Topics/Contents

Unit-IV Bayesian Networks 04 Hours

Introduction of Bayesian Nets, Bayesian Networks- Directed Acyclic Graphs, Bayesian Networks and Proof of Theorem, Bipartite Multiple-Diagnosis Problems, Singly Connected DAGs, Deduction, Abduction, and Induction, Multiple Diagnosis in Bipartite Networks, Irredundant Covers, An Algorithm for Singly Connected Networks, Some Theorems and Proofs - Two Theorems, Formulas for PI and Lambda, Certainty Factors, What Are Certainty Factors? An Interpretation of Certainty Factors, Limits on Certainty

Unit-V Fuzziness and Belief Theory 04 Hours

Fuzziness, The Fuzzy Set Concept, Properties of Fuzzy Sets, Fuzzy Predicates, Fuzzy Rule Systems, Dempster-Shafer Belief Theory, Combining Independent Evidence, Looking Backward - Long Chains of Deductions, Robustness, Computational Feasibility, Psychological Validity, Choosing a System

Unit- Minimization and Feedforward Nets 04 Hours

Findings Minima - Types of Algorithms - Newton's Method, Linear Methods, Quadratic Methods, Introduction to Perceptrons, Algorithm for perceptron, Backpropagation for Feedforward Nets, Assumptions and Notation, Backpropagation, Parameter Issues in Feedforward Nets, Activation Functions, Initialization, Excessive Parameter Growth and Stopping

Assessment Scheme (2 Credit Theory Course): 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.

Assessment scheme covers following aspects of Modified Blooms Taxonomy:
L2 Understanding, L3 Apply, L3 Design, L3 Implement, L4 Analyze and L5 Evaluate

Class Test-1- In 6th Week, based on unit 1 and unit 2, 35 Marks converted to 35 equivalent Marks
Class Test-2- In 12th Week, based on unit 3 and unit 4, 35 Marks converted to 35 equivalent Marks
Comprehensive Viva Voce- End Semester Assessment based on all Units 100 Marks converted to 30 equivalents Mark

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

1. *Edward A. Bender, Mathematical Methods in Artificial Intelligence, IEEE Computer Society Press Los Alamitos, California,* SBN: 9780818672002, 9780818672002
2. *Nelson, H , Essential Math for AI, ISBN 9781098107581, O'Reilly Media, January 2023*

Course Outcomes:

The student will be able to –

1. Understand use of prepositional logic, predicate logic to reason about the relationships between propositions and predicates.
2. Select appropriate nonmonotonic reasoning approach to capture and represent defeasible inferences
3. Apply different trees and search algorithms to find the best possible solution inferences
4. Comprehend and use Bayesian networks to solve a problem which has uncertainty due to multiple events
5. Apply Fuzzy logic use to imitate human reasoning and cognition
6. Find minimum to for the purpose of optimization

CO-PO Map:

	Program Outcomes (PO)												**PSO**			
	PO 1	**PO2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO8**	**PO 9**	**PO1 0**	**PO1 1**	**PO1 2**	**PSO 1**	**PSO 2**	**PSO 3**	**PSO 4**
1	2	3	3	2								2	3			
2	2	3	3	2								2	3			
3	2	3	3	2								2	3			
4	2	3	3	2								2	3			
5	2	3	3	2								2	3			
6	2	3	3	2								2	3			
Avg	**2**	**3**	**3**	**2**								**2**	**3.0**			

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L2, CO3 – L3, CO4 – L4, CO5 – L4 and CO6 – L5

Syllabus Template **FF No.: 654**

CI2006: Design Thinking 3 and 5

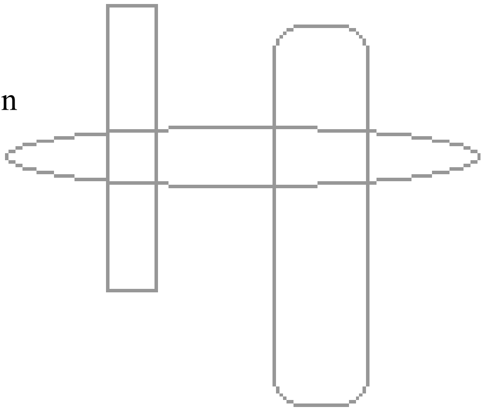
Credits: 01 **Teaching Scheme: Tutorial 01 Hr/week**

Course Prerequisites: Problem Based Learning, Project Centric Learning

Course Objective:
To provide ecosystem for students and faculty for paper publication and patent filing

Section 1: Topics/Contents

What is research?
Importance of Paper Publication and Patents
Structure of Paper
Journal Publication
Publication in conference
Literature Review
Research Paper Writing
Journal Ratings and Evaluation
How to rate a Journal?
Intellectual property (IP)
Research Ethics
Entrepreneurship



Section 2: Topics/Contents

Structure of the paper
Journal List (Top 50 Journals)
Selection of the journal
Use of various online journal selection tools
Plagiarism checking
Improving contents of the paper
Patent drafting
Patent search
Filing of patent
Writing answers to reviewer questions
Modification in manuscript
Checking of publication draft

Course Outcome: [Publication of paper or patent]

The student will be able to

1. Understand the importance of doing Research
2. Interpret and distinguish different fundamental terms related to Research
3. Apply the methodology of doing research and mode of its publication
4. Write a Research Paper based on project work
5. Understand Intellectual property rights
6. Use the concepts of Ethics in Research
7. Understand the Entrepreneurship and Business Planning

CO-PO Mapping:

	Program Outcomes (PO)												PSO			
CO/PO	1	2	3	4	5	6	7	8	9	10	11	12	PSO 1	PSO 2	PSO 3	
CO1	1	1	1	1	1	--	--	--	--	--	--	1	1	2	2	
CO2	1	1	1	1	1	--	--	--	--	--	--	1	1	2	2	
CO3	2	2	3	3	2	2	1	2	2	3	--	1	1	3	3	
CO4	3	3	3	3	3	2	1	2	2	3	1	1	1	3	3	
CO5	1	1	1	1	1	--	--	--	--	--	--	1	1	2	2	
CO6	2	2	2	2	2	2	1	3	2	3	--	1	1	3	3	
C07	1	1	1	1	1	--	--	--	--	--	--	1	1	1	1	
Average	1.57	1.57	1.71	1.71	1.57	2.0	1.0	2.33	2.0	3.0	1.0	1.0	1.0	2.28	2.28	

Syllabus Template **FF No. : 654**

CI2007: Engineering Design and Innovations-III

Credits:.4 **Lab: 8 hours/Week**

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

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.

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, Defence, 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

EDD 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.

3. 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:

	Program Outcomes (PO)												PSO			
CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	
CO1	2	2	2	2					3		2	2	2	3	3	
CO2	2	2	3	2	2		2		3		2	2	2	3	3	
CO3	2	2	3	2	3		2		3		2	2	2	3	3	
CO4	2	2	3	2	3	3		2	3		2	2	2	3	3	
CO5	2	2	3	2	3	2			3		2	2	2	3	3	
CO6	2	2	3	3	2				3		3	2	3	3	3	
Average	2.0	2.0	2.83	2.83	2.6	2.5	2.0	2.0	3.0	1.0	2.16	2.0	2.17	3.0	3.0	

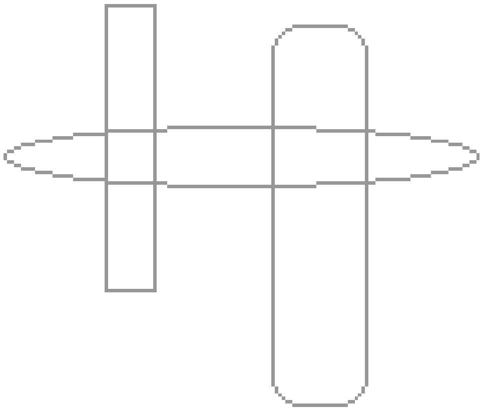
CO attainment levels

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

S. Y. B. Tech. Computer Science & Engineering (Artificial Intelligence)

AY 2024-25

Module IV Course Content



Syllabus Template **FF No.: 654**

CI2001: Data Structures

Credits: 4....... **Teaching Scheme Theory: 2 Hours/Week**
Tutorial: 1 Hours/Week
Lab: 2 Hours/Week

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

Course Objectives:

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

2. To learn and understand linear and non-linear data structure constructs.

3. To implement searching and sorting techniques using linear data structures.

3. To understand how to solve problems using step by step approach with the help of fundamental data structures.

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

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 1: Topics/Contents

Unit-I: Arrays: **(5 Hours)**

Memory Representation and application of Single and Multidimensional arrays, Sparse Matrix. **Searching and Sorting techniques:** Linear Search, Binary search with Analysis. **Sorting Techniques:** Bubble Sort, Insertion Sort, Merge Sort, Quick Sort with Analysis and passes.

Unit-II: Linked Lists: **(5 Hours)**

Dynamic memory allocation, Singly Linked Lists, Doubly linked Lists, Circular linked lists and Generalized linked lists, Applications of Linked list, introduction to Vectors and Application.

Unit-III: Stack: **(4 Hours)**

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.

Section2: Topics/Contents

Unit-IV: Trees: (5.. Hours)

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, Heap Tree and Heap Sort, Huffman Tree.

Unit V: Graphs: (5..Hours)

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.

Unit VI: Hashing: (4..Hours)

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

List of Tutorials (13):

1) Sorting Techniques: Selection Sort, Shell Sort, Radix Sort.
2) Generalized Linked Lists.
3) Problem solving using stack (Maze problem, Tower of Hanoi).
4) Expression conversion like infix to prefix and postfix and vice versa.
5) Priority Queues and Job Scheduling Algorithm.
6) Threaded Binary tree and Stack less Traversals using TBT.
7) AVL Tree.
8) Red-Black Tree.
9) B-Tree.
10) B+ Tree.
11) Applications of Graph in Network problems.
12) Searching Techniques: Ternary Search, Fibonacci Search.
13) Design of Hashing Functions and Collision Resolution techniques.

List of Practical's (Minimum Six):

1) To implement the sorting algorithms.

2) To implement the linked list.

3) To implement any application of Stack data structure.

4) To implement any application of Queue data structure.

5) To implement an algorithm to perform Binary Search Tree (BST) operations (Create, Insert, Delete and Traversals).

6) To implement an algorithm to perform various operations on Binary Tree (Mirror image, Height, Leaf node display, Level wise display etc.)

7) To implement an algorithm to perform various Tree traversals using Stack.

8) To implement a Threaded Binary Tree (TBT).

9) To implement Graph traversal: algorithms: Depth First Search and Breadth First Search.

10) To implement Prim’s and Kruskals Algorithms to find a Minimum Spanning Tree (MST).

11) To implement Dijkstra’s algorithm to solve a Single Source Shortest Path Problem.

12) To implement Hashing algorithms.

List of Course Project areas:

1) Finding Nearest Neighbors.

2) Calendar Application using File handling.

3) Path finder in Maze.

4) Word Completion Using Trie.

5) Bloom Filters.

6) Different Management Systems.

7) Scheduling Applications and Simulation.

8) Shortest Path Applications. (Kirchhoff’s Circuit, TSP with Scenarios).

9) Efficient Storage and Data Retrieval Systems.

10) Different Gaming Application.

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.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks

Course Project: End Semester Examination: 100 Marks converted to 20 Marks

Laboratory Practical: End Semester Examination: 100 Marks converted to 50 Marks

Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

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

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

1. J. Tremblay, P. Soresan, “An Introduction to data Structures with applications”, TMHPublication, 2nd Edition.

MOOCs Links and additional reading material:

1. www.nptelvideos.in,
2. www.geeksforgeeks.org
3. https://www.youtube.com/watch?v=244YpoG1pqA&list=PLrikLQMZHuSonRoDheibeb9ffd9phWIyu&index=5
4. https://classroom.volp.in/

Course Outcomes:

The student will be able to –

1) Choose and make use of single and multi-dimensional array for searching and sorting based applications.

2) Construct computer science applications with the help of dynamic storage representation.

3) Build computer science applications using stacks and queues.

4) Demonstrate the use of tree data structure to represent and manipulate hierarchically organized data in various applications.

5) Utilize graph data structure to design social media, network based and circuit applications.

6) Design and develop the single and multithreads applications by applying hash table and hash map techniques.

CO-PO Map:

	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CS2202.1	2	2	3									2	3		2	
CS2202.2	2	3	2									2	3		2	
CS2202.3	3	3	3				2	2				2	3	2	2	2
CS2202.4	3	3	3	3			2	2				2	3	2	2	2
CS2202.5	3	3	2									2	3		2	
CS2202.6	2	3	3									2	3		2	
Average	2.5	3.0	2.66	3.0			2.0	2.0				2.0	3.0	2.0	2.0	2.0

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

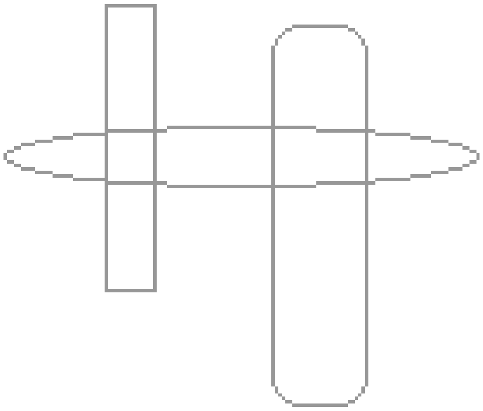
CO1 – L3, CO2– L3, CO3 – L2, CO4 – L4, CO5 – L4 and CO6 – L5

Future Course Mapping:

Design and Analysis of Algorithms, Operating Systems, Compiler Design, Systems Programming, Data Science and similar courses.

Job Mapping:

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



Syllabus Template

FF No. : 654

CI2002: Artificial Intelligence

Credits: 4....... **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites:

- A course on "Computer Programming and Data Structures"
- A course on "Mathematical Foundations of Computer Science"
- Some background in linear algebra, data structures and algorithms, and probability will be helpful

Course Objectives:

1. To learn the distinction between optimal reasoning Vs. human like reasoning
2. To understand the concepts of state space representation, exhaustive search, heuristic search together with the time and space complexities.
3. To learn different knowledge representation techniques.
4. To understand the applications of AI, namely game playing, theorem proving, and machine learning.

Course Relevance: Technologies driven by artificial intelligence (AI) have transformed industries and everyday life. The possibilities for AI applications are virtually unlimited and sought after in practically every industry segment. That's why global organizations are actively recruiting professionals with specialized skills and proficiencies needed to develop future AI technological innovations.

SECTION-I -Topics and Contents:

Unit-I Title: Fundamentals of Artificial Intelligence

Introduction: A.I. Representation, Non-AI & AI Techniques, Representation of Knowledge, Knowledge Base Systems, State Space Search, Production Systems, Problem Characteristics, Types of production systems, Turing Test. **Intelligent Agents**: Agents and Environments, concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation. **Formulation of problems**: Vacuum world, 8 queens, Route finding, robot navigation.

Unit-II Title: UninformedSearch Strategies

Uninformed Search Methods: Depth First Search, Breadth First Search, Depth Limited Search, Iterative Deepening Depth First Search, Bidirectional Search, Comparison of Uninformed search Strategies.

Unit-III Title: Informed Search Methods:

Generate & test, Hill Climbing, Best First Search, A* and AO* Algorithm, Constraint satisfaction, Means Ends Analysis, **Game playing:** Minimax Search, Alpha-Beta Cut offs, Waiting for Quiescence

SECTION-II Topics and Contents:

Unit-IV Title: Logical Agents:

Knowledge based agents, Wumpus world. **Propositional Logic**: Representation, Inference, Reasoning Patterns, Resolution, Forward and Backward Chaining. **First order Logic**: Representation, Inference, Reasoning Patterns, Resolution, Forward and Backward Chaining.

Unit-V Title: Basics of PROLOG:

Representation, Structure, Backtracking. **Expert System**: Design, Implementation, Case study of Expert System in PROLOG

Unit-VI Title: Planning:

Blocks world, STRIPS, Implementation using goal stack, **Planning with state space search:** Forward state space search, Backward state space search, Heuristics for state space search. Partial Order Planning, Planning Graphs, Hierarchical planning, Least commitment strategy. Conditional Planning, Continuous Planning

List of Tutorials

1. AI problem formulation
2. Task Environment

3. AI Problem Characteristics
4. Missionaries and Cannibals Problem
5. Water Jug Problem
6. Monkey Banana problem
7. 8 Puzzle Problem
8. Magic Square problem
9. Tic-Tac Toe Problem
10. Robot Navigation
11. Propositional Logic Examples
12. Predicate Logic Examples
13. Mini Expert system examples

List of Practical's

1. Implementation of AI and Non-AI technique by implementing any two player game
2. Implementation of Uninformed strategies
3. Implementation of Informed strategies
4. Implementation of CSP Problem
5. Implementation predicate logic using PROLOG
6. Implementation of Expert system using PROLOG

List of Course Project Topics (Sample topics)

1. Inventory management E Commerce
2. stock market price prediction
3. Object 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. Fundamentals of Artificial Intelligence

2. Intelligent Agents
3. Uninformed searching Techniques
4. Informed searching Techniques
5. Gaming Techniques
6. Planning Techniques
7. Applications of AI
8. Predicate Logic
9. Propositional Logic
10. Adversarial Search Techniques

List of Home Assignments:

List of Design Based Home Assignments

1. Design of intelligent algorithm for AI Accessibility [CO3] [PO3]
2. Design of AI algorithm for Robot Navigation. [CO3] [PO3]
3. Design of AI algorithm for Customer Experience [CO3] [PO3]
4. Design of AI algorithm for Data-Informed Design [CO3] [PO3]
5. Design of AI algorithm for AI Decision Making [CO3] [PO3]
6. Design of AI algorithm for any application for Children [CO3] [PO3]
7. Design of AI algorithm for problems of Senior Citizens [CO3] [PO3]
8. Design of AI algorithm for ecommerce Applications [CO3] [PO3]
9. Design of AI algorithm for Enterprise UX Design [CO3] [PO3]
10. Design of AI algorithm as Teaching Aid for teachers [CO3] [PO3]

List of Case Study Based Home Assignments

1. How Automobile Sector 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
6. Challenges/Issues in AI applications
7. Research problems in AI
8. AI in Search Engine
9. Future of AI
10. AI in Agriculture

List of Blog Based Home Assignment

1. AI Trends

2. AI Research
3. AI Chatbot
4. Chatbot Magazine
5. AI Medical / Agriculture
6. AI Challenges
7. Knowledge based Inference Engine
8. Rule based inference Engine
9. Truth maintenance system
10. AI in CSP problems

List of Survey Based Home Assignments

1. Adaption of AI in 2020
2. AI in Industry
3. AI in Digital Marketing
4. AI in Gaming
5. AI after Covid-19
6. AI in rule-based systems
7. Analysis of Search Engines: AI perspective
8. Page rank algorithms in AI
9. AI in Ecommerce
10. Analysis of Expert systems in medical diagnosis

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Home Assignments: 100 Marks converted to 20 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (Written): 60 Marks converted to 30 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

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

Course Outcomes:

On the completion of course, student will able to

1. Understand the basics of the theory and practice of Artificial Intelligence as a discipline and about intelligent agents capable of problem formulation.
2. Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.
3. Evaluation of different uninformed and informed search algorithms on well formulated problems along with stating valid conclusions that the evaluation supports.
4. Formulate and solve a given problem using Propositional and First order logic.
5. Analyze the AI problem using different planning techniques.
6. Design and carry out an empirical evaluation of different algorithms on problem formalization, and state the conclusions that the evaluation supports.

CO-PO Map:

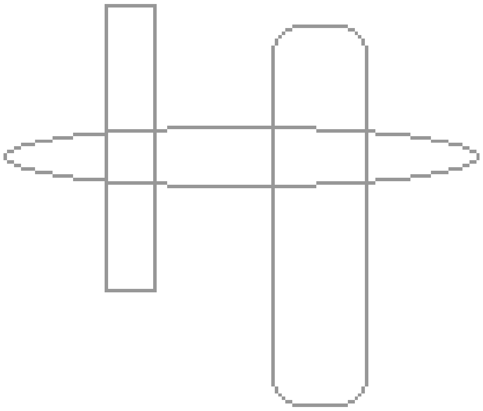
CO	Program Outcomes (PO)												PSO			
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	
CS3226.1	3											2	3	2	2	
CS3226.2	3	2		1								2	3	2	2	
CS3226.3	3	3		2								2	3	2	2	
CS3226.4	3	3	3	2								2	3	3	2	
CS3226.5	3	3		2								2	3	3	2	
CS3226.6	3	3	3	2			3	2	2	2		3	3	3	2	
Average	3.0	2.8	3.0	1.8			3.00	2.00	2.00	2.00		2.16	3.0	2.5	2.0	

CO attainment levels:

CO1 -2 CO2 -2 CO3 -3 CO4-1 CO5-1 CO6-1

Future Course Mapping: *Machine Learning*

Job Mapping: *AI Data Analyst, Data Scientist*



Syllabus Template **FF No. : 654**

CI2003: Operating Systems

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites: Computer Architecture & organization, Data Structure

Course Objectives:

1.To learn functions of Operating System

2.To learn the importance of concurrency and how to implement concurrent abstractions correctly in an OS.

3.To learn OS scheduling policies and mechanisms.

4.To deal with deadlock

5.To learn memory management schemes in various ways to improve performance, and how this impacts system complexity

6. To learn design & develop the Operating system from a scratch.

Course Relevance:

Section 1: Topics/Contents

Unit-I Introduction to OS [04 Hours]

What is OS, Interaction of OS and hardware, Goals of OS, Basic functions of OS, OS Services, System Calls, Types of system calls.

Types of OS: Batch, Multiprogramming, Time sharing, Parallel, Distributed & Real-time OS.

Unit-II Title: Process Management [06 Hours]

Process Concept, Process States: 2, 5, 7 state models, Process Description, Process Control.

Threads: Multithreading models, Thread implementations – user level and kernel level threads, Symmetric Multiprocessing.

Concurrency: Issues with concurrency, Principles of Concurrency

Mutual Exclusion: H/W approaches, S/W approach, OS/Programming Language support: Semaphores, Mutex and Monitors.

Classical Problems of Synchronization: Readers-Writers problem, Producer Consumer problem, Dining Philosopher problem

Unit-III Title: Process Scheduling [04 Hours]

Uniprocessor Scheduling: Scheduling Criteria, Types of Scheduling: Preemptive, Non-preemptive, Long-term, Medium-term, Short-term.

Scheduling Algorithms: FCFS, SJF, RR, Priority.

Section 2: Topics/Contents

Unit-IV Title: Deadlocks [04 Hours]

Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Deadlock Recovery

Unit-V Title: Memory Management [06 Hours]

Memory Management concepts: Memory Management requirements, Memory Partitioning: Fixed, Dynamic Partitioning, Buddy Systems, Fragmentation, Paging, Segmentation, Address translation.

Placement Strategies: First Fit, Best Fit, Next Fit and Worst Fit.

Virtual Memory: Concepts, Swapping, VM with Paging, Page Table Structure, Inverted Page Table, Translation Lookaside Buffer, Page Size, VM with Segmentation with combined paging and segmentation.

Page Replacement Policies: FIFO, LRU, Optimal, Clock.

Swapping issues: Thrashing

Unit-VI Title: I/O and File Management [04 Hours]

I/O management: I/O Devices - Types, Characteristics of devices, OS design issues for I/O management, I/O Buffering.

Disk Scheduling: FCFS, SCAN, C-SCAN, SSTF.

File Management: Concepts, File Organization, File Directories, File Sharing. Record Blocking, Secondary Storage Management, Free Space management, Security.

List of Tutorials (13):

1. Linux Commands
2. Shell Script
3. Types of OS
4. System Call
5. Synchronization Problems
6. Problems on CPU Scheduling
7. Problems of Deadlock detection, avoidance
8. Problems on Placement Strategies
9. Problems on page Replacement Algorithms
10. Problems on address translation in paging & segmentation
11. Problems on Disk scheduling algorithm
12. iOS -Case Study
13. Android OS -Case Study

List of Practical's (Minimum Six to be performed out of 10):

1. Execution of Basic & Advanced Linux Commands.
2. Write shell script covering – basic arithmetic, control structures, loops, execution of Linux command in shell, command line arguments, functions and arrays.
3. Solve synchronization problems – Reader writer problem, Producer consumer problem & dinning philosopher problem using mutex & semaphore.
4. Implement CPU scheduling algorithms
5. Implement Banker's algorithm
6. Implement deadlock detection algorithm
7. Implement placement strategies.
8. Implement buddy system.
9. Implement page replacement algorithm
10. Implement disk scheduling algorithm

List of Course Project areas:

1. Design and implementation of a

i. CPU/ Machine Simulation

ii. Supervisor Call through interrupt

Design multi programming operating system phase 1

2.Design and implementation of a Multiprogramming Operating System: Stage II

i. Paging

ii. Error Handling

iii. Interrupt Generation and Servicing

iv. Process Data Structure

3.Design and implementation of a Multiprogramming Operating System: Stage III

i. I/O Channels & I/O buffering

ii. Multiprogramming

iii. I/O Spooling

4. Design multi programming operating system phase 1 with arithmetic & logical instructions

5.Design multi programming operating system phase 3 with swapping

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.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Laboratory Practical: End Semester Examination: 100 Marks converted to 50 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

Text Books: *(As per IEEE format)*

1. *Stalling William; "Operating Systems", 6thEdition, Pearson Education.*
2. *Silberschatz A., Galvin P., Gagne G.; "Operating System Concepts", 9th Edition, John Wiley and Sons.*
3. *D M Dhamdhere; "Systems Programming & Operating Systems"; Tata McGraw Hill Publications, ISBN – 0074635794*
4. *John J Donovan; " Systems Programming " ; Tata Mc-Graw Hill edition , ISBN-13978-0-07-460482-3*

Reference Books: *(As per IEEE format)*

1. *Silberschatz A., Galvin P., Gagne G ;"Operating System Principles" 7th Edition John Wiley and Sons.*
2. *YashavantKanetkar; "Unix Shell Programming", 2 nd Edition, BPB Publications.*
3. *Forouzan B. A., Gilberg R. F.; "Unix And Shell Programming", 1 st Edition, Australia Thomson Brooks Cole.*
4. *Achyut S. Godbole ,AtulKahate; "Operating Systems", 3 rd Edition, McGraw Hill.*

MOOCs Links and additional reading material:

1. https://nptel.ac.in/courses/106105214
2. https://onlinecourses.nptel.ac.in/noc20_cs04/preview
3. https://archive.nptel.ac.in/courses/106/102/106102132/
4. https://onlinecourses.nptel.ac.in/noc21_cs72/preview

Course Outcomes:

The student will be able to –

1. Discuss the functions of a contemporary Operating system with respect to convenience, efficiency and the ability to evolve.
2. Implement concurrent abstractions correctly in an OS to solve real world problems.
3. Use various CPU scheduling algorithms to construct solutions to real world problems.
4. Corelate the mechanisms related to deadlock handling in real life situations.
5. Distinguish memory management schemes & file management systems in various ways to improve performance, and analyze the impact of it on system complexity.
6. Design & develop the Operating system from a scratch

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	
1	2	2											3	2		
2	2	2	3	2			2	2					3	3		
3	2	3	3	2			2	2					3	2		
4	2	3		2									3	2		
5	2	3		1									3	2		
6	2	2	3	2		2	2	2	3	3.0	3.0	3.0	3	3		
Avg	2	2.5	3.0	1.8		2.0	2.0	2.0	3.0	3.0	3.0	3.0	3.0	2.33		

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

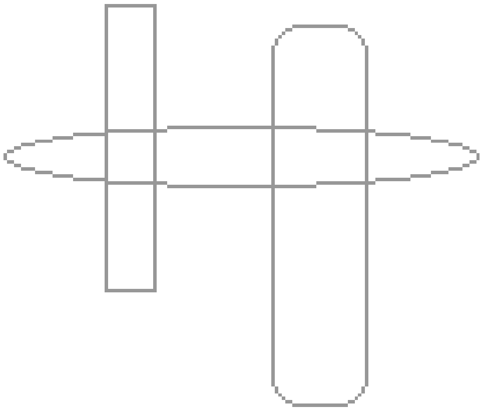
CO1 – L1, CO2 – L2, CO3 – L3, CO4 – L4, CO5 – L4 and CO6 – L5

Future Course Mapping:

Advanced Operating System, Distributed Operating System, Parallel Computing.

Job Mapping:

System Administrator, System Analyst



CI2004: Data Visualization

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites: Fundamentals of Python Programming

Course Objectives:

1. Identify the different types of data and create data visualization
2. Transform raw data into understandable format and conduct exploratory analysis
3. Use python and R libraries for data preprocessing and visualization
4. Use knowledge of perception and cognition to evaluate visualization design alternatives.
5. Design and evaluate results of exploratory data analysis.
6. Apply data transformation and paraphrase the results for documentation.

Course Relevance: Data visualization is the graphical representation of data and information using visual elements such as charts, graphs, maps, and infographics. It involves the creation of visual representations that help people understand and interpret complex data sets more easily. Data visualization is widely used across various industries and domains to communicate data-driven insights, patterns, and trends effectively.

Section 1: Topics/Contents

Unit-I Introduction and Describing Data **[04 Hours]**
Importance of analytics and visualization in the era of data abundance. Review of probability, statistics and random processes. -Brief introduction to estimation theory. Sources of Data, Process for Making Sense of Data, Observations and Variable, Types of Variables, Central Tendency, Distribution of the Data, Need of Data wrangling Methods, Confidence Intervals, Hypothesis Tests

Unit-II Data Manipulation **[06 Hours]**
Installing and using Pandas, Introduction of Pandas Objects, Data Indexing and Selection, Operating on Data in Pandas, Handling Missing Data, Hierarchical Indexing, Combining Datasets: Concatenation and Append, Combining Datasets: Merge and Join, Aggregation and Grouping, Pivot Tables, Vectorized String Operations, Working with Time Series, High-Performance Pandas.

Unit-III Data Visualization [04 Hours]

Visualization Design Principles, Tables, Univariate Data Visualization, Multivariate Data Visualization, Visualizing Groups, Dynamic Techniques General Matplotlib Tips, Two Interfaces for the Price of One, Simple Line Plots, Visualizing Errors, Density and Contour Plots, Histograms, Binning, and Density, Customizing Plot Legends, Customizing Color bars, Multiple Subplots, Text and Annotation, Customizing Matplotlib.

Section 2: Topics/Contents

Unit- IV Visualization Using Seaborn [06 Hours]

Introduction to Seaborn: Seaborn functionalities and usage, Spatial Visualizations and Analysis in Python, R Programming: Bar plot, Plotting categorical data, Stacked bar plot, Histogram, plot () function and line plot, Pie chart / 3D pie chart, Scatter plot, Box plot.

Unit-V Information visualization [04 Hours]

Clustering Techniques, Dimension reduction, PP, MDS, graph visualization techniques for big data, visual analytics, Statistical methods, Information theory for big data visualization.

Unit-VI Applications of Data Visualizations [04 Hours]

Business Intelligence, Data Exploration and Analysis, Presentations and Reporting, Geographic Information Systems (GIS), Healthcare and Medicine, E-commerce and Customer Analytics.

List of Tutorials (13): [CO’s Mapped]

1. Study installation and configuration of Tableau
2. Data Visualization using statistical inference methods
3. Hypothesis Testing: Null and Alternative
4. Predictive data Visualization with Python
5. Clustering and Time-series analysis using Scikit- learn sklearn, metrics, Confusion matrix, AUC-ROC Curves, Elbow plot
6. Linear Regression
7. Data visualization using ggplot
8. Scala
9. Dashboard
10. Applications

List of Practical’s (Minimum Six):[CO’s Mapped]

Use data set of your choice from Open Data Portal (https://data.gov.in/) for the following exercises.

1. Study installation and configuration of R programming

2. Data Wrangling, IN Perform the following operations using Python on any open source dataset (e.g., data.csv)
   - *a.* Import all the required Python Libraries.
   - *b.* Locate open-source data from the web (e.g., https://www.kaggle.com). Provide a clear description of the data and its source (i.e., URL of the web site).
   - *c.* Load the Dataset into pandas data frame.
   - *d.* Data Preprocessing: check for missing values in the data using pandas isnull(), describe() function to get some initial statistics. Provide variable descriptions. Types of variables etc. Check the dimensions of the data frame.
   - *e.* Data Formatting and Data Normalization: Summarize the types of variables by checking the data types (i.e., character, numeric, integer, factor, and logical) of the variables in the data set. If variables are not in the correct data type, apply proper type conversions.
   - f. Turn categorical variables into quantitative variables in Python. Practical based on Data Loading, Storage and File Formats
3. Descriptive Statistics - Measures of Central Tendency and variability perform the following operations on any open-source dataset (e.g., data.csv) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variable. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable.
4. Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris-versicolor' of iris.csv dataset.
5. Implement logistic regression using Python/R to perform classification on Social_Network_Ads.csv dataset.
6. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.
7. Write a code to check how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram.
8. Use the inbuilt dataset 'titanic' as used in the above problem. Plot a box plot for distribution of age with respect to each gender along with the information about whether they survived or not. (Column names: 'sex' and 'age')
9. Write observations on the inference from the above statistics.
10. Download the Iris flower dataset or any other dataset into a Data Frame. (e.g., https://archive.ics.uci.edu/ml/datasets/Iris). Scan the dataset and give the inference as:
    - a. List down the features and their types (e.g., numeric, nominal) available in the dataset.
    - b. Create a histogram for each feature in the dataset to illustrate the feature distributions.
    - c. Create a boxplot for each feature in the dataset.
    - d. Compare distributions and identify outliers.

11. Write an R script to do the following:
    - a. Simulate a sample of 100 random data points from a normal distribution with mean 100 and standard deviation 5 and store the result in a vector.
    - b. Visualize the vector created above using different plots.
    - c. Test the hypothesis that the mean equals 100
    - d. Use wilcox test to test the hypothesis that mean equals 90.

List of Course Project Topics:

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. Fraud detection
10. Visualization techniques such as Chernoff's faces.

List of Group Discussion Topics:

1. scanpy
2. Tools to Design or Visualize Architecture of Neural Network
3. Plexus-Interactive Emotion Visualization based on social media
4. Pre-built Grafana Dashboard
5. Single cell interactive plotting tools
6. Interactive word cloud
7. Encoding Data with Marks and Channels
8. Rendering Marks and Channels with D3.js and SVG
9. Introduction to D3 Scales
10. Online data visualization

List of Seminar Topics:

1. Bring data to life with SVG,Canvas and HTML
2. Progressive <svg> pie, donut, bar and line charts
3. A visualization grammar
4. Matplotlib: plotting with Python
5. Python svg Graph plotting Library
6. A general purpose, real-time visualization library
7. An object-oriented API for business analytics
8. Air pollution visualization and forecasting platform based on Spring Cloud
9. A time visualization of keywords in Twitter with Python and tweepy, json, panda, and Vincent libraries
10. Agriculture Commodities Prices Seasons

List of Design based Home Assignments:

1. A program for visualizing data from a Whats App chat.
2. Repository for the tidy Nano package to analyze and visualize Nano Sight/NTAdata
3. Wrangling, Exploring, Analyzing, and Visualizing Data
4. Visualizing discrete and continuous data for known groups
5. GUI for displaying Binary Search Trees written in C++ using the QtLibrary.

List of Case Study based Home Assignments:

1. Hub for data visualization & web contents using Java Script different chart library
2. Plexus –Interactive Emotion Visualization based on social media
3. Useful form a king plots from MEICA
4. Interactive county-level mapping project for United States
5. A collection of React components for building interactive data visualizations

List of Blog based Home Assignments:

1. Retrieving, Processing, and Visualizing Data with Python
2. Meta Graph: a work bench for interactive exploratory data analysis of large expression datasets
3. Map-of-emergency-incidents
4. A small data visualization for Assembly Election results in various Indian states
5. GDD: Growing Degree Days Visualization

List of Survey based Home Assignments:

1. Visualization for analyzing Machine Learning & feature engineering
2. Google Me
3. Visualizing survey data
4. Visualizing Real Time data
5. Visualizing Big data

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Presentation: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (Written): 60 Marks converted to 30 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

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

1. Mario Dobler, Data Visualization with Python: Create an impact with meaningful data insights using interactive and engaging visuals, February 2019, Packt Publishing
2. Kirthi Raman, Mastering Python Data Visualization Paperback, Publishing
3. Gardener M, Beginning R: The statistical programming language, WILEY. (2017).
4. Lawrence, M., & Verzani, J. Programming Graphical User Interfaces in R. CRC press. (ebook)

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

1. Kieran Healy Chen, Data Visualization– A Practical Introduction Paper back–Import, 4 Jan 2019.
2. Chun-houh, Wolfgang Karl Härdle, and Antony Un win, Eds, Handbook of data visualization. Springer Science & Business Media.
3. Cotton, R., Learning R: a step-by-step function guide to data analysis, O'reilly Media Inc.

MOOCs Links and additional reading material:

1. https://elearn.nptel.ac.in/shop/iit-workshops/completed/data-visualization-with-r/
2. https://www.coursera.org/learn/datavisualization

Course Outcomes:

The student will be able to –

1. Understanding fundamental principles and concepts behind effective data visualization.
2. Design and classify different data visualization techniques.
3. Analyze exploratory data analysis using visualization.
4. Design and evaluate Python and R programming libraries for data visualization
5. Evaluate and apply data transformations Techniques.
6. Develop a visualization intensive project.

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO1	PSO 2	PSO 3	
CO1	2			2									3	2	3	
CO2	2	2		2							3	3	3	2	3	
CO3	3	3	2		2	3							3	2	3	
CO4	3	3	2	1	2	3	3	3				3	3	2	3	
CO5	3		2	3									3	2	3	
CO6	3		2						3		3		3	2	3	
Averag e	3	3	1	1	1	1	1	1	1	1	0	1	3.0	2.0	3.0	

CO Attainment levels:

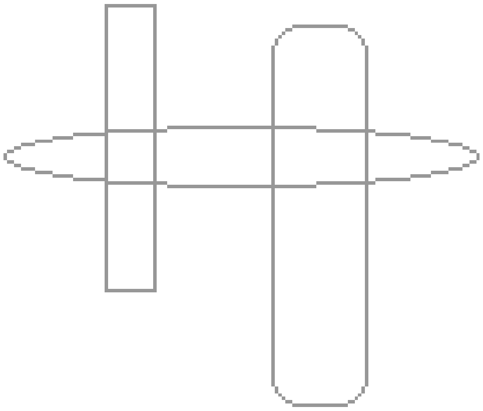
Weights for attainment levels: L1 - Easy-0.75
L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L2, CO3 – L3, CO4 – L4, CO5 – L4 and CO6 – L5

Future Course Mapping:
Mention other courses that can be taken after completion of this course:

Database Management Systems, Data Analytics, Software Engineering, Machine Learning, Deep Learning, Fuzzy Logic.

Job Mapping:
Job opportunities that one can get after learning this course are as:
Application developer, power BI developer-data visualization, data visualization engineer, data visualization expert, data analytics, specialist-visualization, software development engineer-data visualization, data visualization designer, senior analyst-visualization, Scientist.



CI2005: Theory of Computation

Credits: 2....... **Teaching Scheme Theory: 2 Hours/Week**

Course Prerequisites: Introduction to discrete mathematics, proof techniques, basic familiarity with programing/computing.

Course Objectives:

1. Students will learn basic concepts such as alphabet, strings, Languages, Decision problems, etc and will be able to work with the abstract formal setup.
2. Students will be able to design deterministic/nondeterministic automata for regular languages, also he will be able to prove non regularity of languages through application of Pumping Lemma and Myhill-Nerode theorem.
3. Students will gain understanding of the role of non-determinism in Automata theory.
4. Students will be able to design Context free grammars, Push down automata for context Free Languages
5. Students will be able to design Turing Machines for various computational problems and see the equivalence of TM model with high level programming languages.
6. Students will be able to comprehend meaning of undecidability in the context of Turing Machine Model and understand the inherent limits of computation.

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 limitation 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/Contents

Unit-I Finite Automata [4 Hours]

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, Applications and Limitation of Finite Automata.

Unit-II Regular and Non-Regular Languages [6 Hours]

Regular expression (RE), Definition, Operators of regular expression and their precedence, Algebraic laws for Regular expressions, Kleene’s Theorem: Equivalence Regular expressions and DFAs (without proof), 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.

Unit-III Context Free Grammars (CFG) [4 Hours]

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. Normal forms for CFGs: CNF and GNF (without proof). Decision Properties of CFLs (Emptiness, Finiteness and Membership). Applications of CFG.

Section2: Topics/Contents

Unit-IV Push Down Automata: [5 Hours]

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, Chomsky hierarchy**.**

Unit-V Turing Machines: [5 Hours]

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.

Unit-VI Introduction to Undecidability: [4 Hours]

Church-Turing Thesis and intuitive notion of Algorithm, 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.

Assessment Scheme (2 Credit Theory Course): 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.

Assessment scheme covers following aspects of Modified Blooms Taxonomy:
L2 Understanding, L3 Apply, L3 Design, L3 Implement, L4 Analyze and L5 Evaluate

Class Test-1- In 6th Week, based on unit 1 and unit 2, 35 Marks converted to 35 equivalent Marks
Class Test-2- In 12th Week, based on unit 3 and unit 4, 35 Marks converted to 35 equivalent Marks
Comprehensive Viva Voce- End Semester Assessment based on all Units 100 Marks converted to 30 equivalents Mark

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

1.Hopcroft J, Motwani R, Ullman, Addison-Wesley, "Introduction to Automata Theory, Languages and 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:

	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	
1	2	3	3	1								2	3	2		
2	2	3	2				2					2	3	2		
3	2	3	2	2			2					2	3	2		
4	2	3							1			2	3	2		
5	2	3	2									2	3	2		
6	2	3		1								2	3	2		
Total	2	3	2.33	1.33	0	0	2	0	1	0	0	2	3	2.0		

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

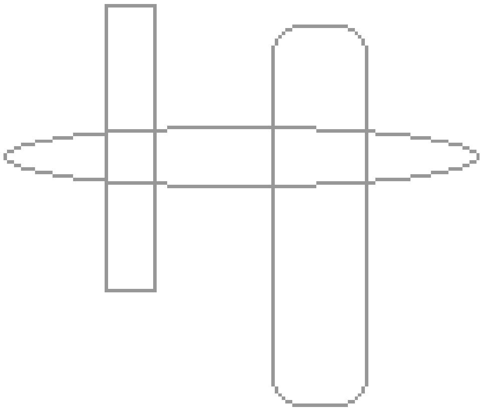
CO1 – L2, CO2 – L3, CO3 – L3, CO4 – L4, CO5 – L5 and CO6 – L5

Future Course Mapping:

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

Job Mapping:

The principles learnt in the course have wide applicability, in domains like Compiler design, Programming languages design, Machine learning, Natural Language processing, etc. Any job that involves modeling and systematic study of some systems, background of Theory of Computation is useful. Understanding of the course content is helpful in developing a systematic and structured approach towards programming. The programming/algorithm design abilities lie at the heart of computer science and are useful for several job profiles in the computer industry. If a student wants to pursue higher education/ research in Computer Science, this course is essential.



T.Y. B. Tech. Computer Science & Engineering (Artificial Intelligence)

AY 2024-25

Module V Course Content

Syllabus Template **FF No.: 654**

CI3001: Computer Network Technology

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**
Tutorial: 1 Hours/Week
Lab: 2 Hours/Week

Course Prerequisites: Operating System, Database Management Systems

Course Objectives:

1. To learn transmission mediums, networking devices and topologies used in the Internet
2. To learn networking standards, IP packet switching and routing used in the Internet
3. To learn transport layer and application layer protocols used in the Internet
4. To learn front end technologies for website development
5. To learn single page applications development using REACT
6. To learn REST API based enterprise website development using REACT, Node JS, Spring Boot with different database technologies

Course Relevance:

The key technology of the information age is communications. Computer network is a truly global area of study, both because the technology enables global communication over telephone lines and the Internet. Computer Networks and 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. Most of the jobs available in the IT industries are web technology related.

Section 1: Topics/Contents

Unit-I Networking Fundamentals and Physical Layer 04 Hours

Network Organizations and Architectures: What is computer Networks, Network Topologies: Mesh, Star and Hierarchical, Types of Computer Networks: LAN, MAN, WAN, PAN, Internet, internet and Intranet. Client-Server; Peer To Peer. Network Architecture Modes: Infrastructure and Ad-hoc mode.

Reference Models: OS and TCP/IP. Design Issues for Layers.

Physical Layer: Transmission Mediums: Air, Vacuum, Cat5, Cat5e, Cat6, Cat6a, Cat7, Cat8, OFC - Single and Multicore.
Networking Devices Wired and Wireless: NIC, Repeater, Bridge, Switch, Modem, Router, Gateways and Access Point.

Unit-II Medium Access Control and Network Layer 04 Hours
Medium Access Control: Legacy Standard : 10 Mbps IEEE 802.3 Standard(Ethernet), High Speed Ethernet Standards: Fast, Gigabit and 10Gigabit.
Wireless Standards: IEEE 802.11a/b/g/n/ac, IEEE 802.15, IEEE 802.15.4 and IEEE 802.16 Standards, CSMA/CA
Switching Techniques and IP Addressing: Circuit, Message and Packet Switching. Logical Addressing: IPv4 and IPv6
Network Layer Protocols: Internet Protocol(IP), Internet Control Message Protocol(ICMP)

Unit-III Transport Layer and Application Layer 05 Hours
Transport Layer Protocols: Transmission Control Protocol (TCP), User Datagram Protocol (UDP)
Services: Berkeley Sockets, Connection Establishment, Connection Release
Application Layer: Domain Name System (DNS) and File Transfer Protocol (FTP)
WWW: Hyper Text Transfer Protocol (HTTP1.1/1.2/2.0) and HTTPS with SSL.
Email: SMTP, MIME, POP3 and Webmail.

<u>Section2: Topics/Contents</u>

Unit-IV Client-Side Technologies 09 Hours HTML5: structure of html document, HTML elements: headings, paragraphs, line break, links, frames, lists, tables, images and forms **CSS3.0:** Styles, colors, fonts and Text Alignments

Java Script: Basics of Document Object Model (DOM), Variable Declarations : Using var, let, and const, Reserved Keywords, Objects and Classes, Understanding Functions: Declarations, Expressions, Arrow Functions, Event Handling- Browser Events and Event Listeners, Form Validation, AJAX
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, Redux

Unit-V Spring Boot 05 Hours
Spring Framework, Spring Boot Framework, Installing Spring Boot, Build Tool Maven/Gradle/Ant, Core Features, Spring Security, Web Applications, JPA for database connectivity, working with SQL

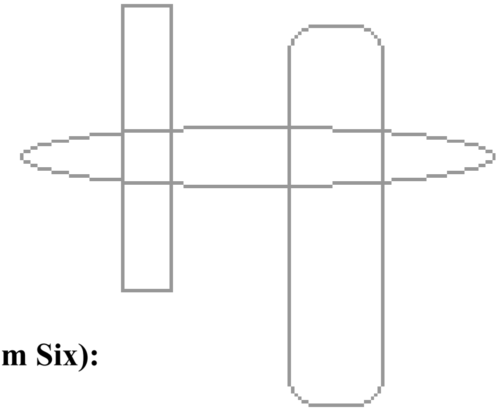
and NoSQL, Messaging, Testing, Deploying Spring Boot Applications, Monitoring and Testing. POSTMAN Tool for API testing.

Unit-VI Node JS **03 Hours**
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 (13):

List of Tutorials (13):

1. Use of XML in web development
2. Use of JSON in web development
3. Learning JQuery
4. Learning JQuery
5. Learning Bootstrap
6. Learning Bootstrap
7. Learning PHP
8. Learning PHP
9. Learning MySQL
10. Learning MySQL
11. Learning Mongo DB
12 Learning Mongo DB
13. Learning REST API



List of Practical's (Minimum Six):

Unit-I and Unit II: Use two turn of lab
1a) Setting up small wired computer network :
Set up a small wired network of 2 to 4 computers using Hub/Switch/. It includes Preparation of Cables and setting up wired network.

1b) Setting up small wireless computer network and hands-on networking command:
Set up a small wired network of 2 to 4 computers using access point and ask students to access it on their wireless gadgets.
Hands on for network commands - ping, pathping, ipconfig/ifconfig, arp, netstat, nbtstat, nslookup, route, traceroute/tracert, nmap.

Unit-II MAC and Network Layer

2) Write a program to find the shortest path using Dijkstra Equation for Link State Routing Protocol which is used by Open Shortest Path First Protocol (OSPF) in the Internet for the network flow provided by instructor.

Unit-III Transport Layer and Application Layers

3a) Write the client server programs using TCP Berkeley socket primitives for wired /wireless network for following
a. to say Hello to Each other
b. File transfer

3b) Write the client server programs using UDP Berkeley socket primitives for wired /wireless network for following
a. to say Hello to Each other
b. Calculator (Trigonometry)

3c) Understanding protocol stack of Intranet
Analyze packet formats of Ethernet, IP, TCP and UDP captured through Wireshark for wired networks.

Unit-IV Client-Side Technologies

4) Design and develop a website using toggleable or dynamic tabs or pills with bootstrap and JQuery to show the relevance of SDP, EDI, DT and Course projects in VIT.

Unit-VSpringboot

5) Design and develop a responsive website to prepare one semester result of VIT students using REACT, Springboot and MySQL/ MongoDB/Oracle. Take any four subjects with MSE Marks (30%) ESE Marks (70%).

Unit-VI NODE JS

6) Design and develop a responsive website for an online book store using REACT, Node JS/ PHP and MySQL/ MongoDB/Oracle having 1) Home Page2) Login Page 3) Catalogue Page: 4) Registration Page: (database)

List of Course Project areas: Networking

1. Design and deploy website for TCP based Multithreaded HTTP client server for accessing student activity data in the institute.
2. Design and deploy website for TCP based Multithreaded FTP client server to share institute level notices.
3. Design and deploy website for UDP based Multithreaded TFTP client server for your class

4. Design and deploy website for TCP based Multithreaded SMTP and POP3 mail client server for your campus.
5. Design and deploy website for TCP based Multithreaded Chat client server for your class.
6. Design and deploy website for UDP based Multithreaded Chat client server for your class.
7. Design and deploy website for UDP based Multithreaded Audio Conferencing client server for computer engineering department.
8. Design and deploy website for UDP based Multithreaded Video Conferencing client server for computer department
9. Design and deploy website to demonstrate implementation of RIP/OSPF/BGP using Packet Tracer
10. Design and deploy website to simulation of AODV routing protocol using Packet Tracer/ NS3/OMNet

List of Course Project areas: Web Technology

1. Develop a responsive web application for Student Grievance System
2. Develop a responsive web application for Workflow Management System for MNC
3. Develop a responsive Gaming Website
4. Develop a responsive web application to help farmers to solve their farming problems
5. Develop a responsive web application for GST Billing Software for Small Business
6. Develop a responsive web application for online Crime Reporting System using PHP
7. Develop a responsive web application for online College Voting System
8. Develop a responsive web application for online Loan Processing System for Farmers.
9. Develop a responsive web application for restaurant food order management
10. Develop a responsive web application for e-book shop
11. Develop a responsive web application for on-line music store
12. Develop a responsive web application for guest visiting management to your society
13. Develop a responsive web application for web search engine

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.
Assessment scheme covers following aspects of Modified Blooms Taxonomy:
L2 Understanding, L3 Apply, L3 Design, L3 Apply, L4 Analyze and L5 Evaluate

Laboratory Continuous Assessment : 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Laboratory: End Semester Examination: 100 Marks converted to 50 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

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

1. Andrew S. Tanenbaum, "Computer Networks",5th Edition, PHI, ISBN 81-203-2175-8.
2. Kurose, Ross "Computer Networking a Top Down Approach Featuring the Internet", Pearson; 6th edition (March 5, 2012), ISBN-10: 0132856204

3. Frouzan B., "Data Communications and Networking", 5th edition, Tata McGraw- Hill, Publications, 2006

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

1. Matthew S. Gast "802.11 Wireless Networks", O'Reilly publications; 2nd Edition.

2. C. Siva Ram Murthy and B. S. Manoj, "Ad Hoc Wireless Networks: Architectures and Protocols" Prentice Hall, 2004

3. Holger Karl and Andreas Willing, "Protocols and Architectures for Wireless Sensor Networks", Wiley, ISBN: 0-470-09510-5

Text Books (Web Technology): *(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 (Web Technology) : *(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:

1. www.w3.org
2. HTML, The Complete Reference
3. www.htmlref.com
4. w3schools.org
5. php.net/ https://jquery.com/
6. developer.mozilla.org/en-US/docs/AJAX
7. www.tutorialspoint.com/css/
8. PHP: Data Structures - Manual -----
9. docs.spring.io/spring-boot/docs/current/reference/html/
10. nodejs.org/en
11. react.dev

MOOCs Links and additional reading material:

www.nptelvideos.in, www.coursera.com, www.udemy.com

Course Outcomes:

The student will be able to –

1. Select topology, essential components of physical layer and networking devices to design computer networks.
2. Build wired and wireless intranet with correct communication and service frameworks.
3. Develop Client-Servers by the means of correct standards, protocols and technologies
4. Build single page applications using REACT as a reusable UI component technology
5. Write Web API/RESTful API application programming interface to communicate with Springboot as a server side technology.
6. **[Group Assignment]** Design and develop three tier enterprise application using client side, server side and back end technologies

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO2	PSO3	
1	3	**3**	2	1	-	2	-	-	1	-	-	1	-	-	-	
2	3	2	**3**	1	3	2	2	2	1	-	-	2	3	3	-	
3	3	3	3	**1**	3	2	-	**2**	1	-	-	2	3	3	-	
4	3	3	3	1	3	**2**	-	2	1	-	-	1	3	3	-	
5	3	2	3	1	3	2	-	2	**1**	-	-	**1**	3	3	-	
6	3	3	3	2	3	3	-	3	3	2	2.0	3	3	**3**	**2**	
Avg	**3**	**2.83**	**3**	**1.16**	**3**	**2.16**	**2**	**2.2**	**1.33**	**2.0**	**2.0**	**1.66**	**3**	**3**	**2**	

Attainment Levels: 2,3,4,3,4, 4

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L3, CO3 – L4, CO4 – L3, CO5 – L4 and CO6 – L4

Future Course Mapping:

High Speed Networks, Wireless Networks, Mobile Networks, Network Security, Cyber Security, Cloud Computing, Distributed System, Mobile Application Development

Job Mapping:

Network Engineer, Network Stack Developers, Application Developer, Software Engineer, Web Developer, IT Engineer, UI Developer

Syllabus Template **FF No. : 654**

CI3002: Design and Analysis of Algorithms

Credits: 4....... **Teaching Scheme Theory: 2 Hours/Week**
Tutorial: 1 Hours/Week
Lab: 2 Hours/Week

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.
5. Students will design randomized, approximation algorithms for some computational problems.
6. Students will be able to incorporate algorithm design principles, data structures and provide efficient solutions for complex computational problems.

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: Topics/Contents

Unit-I Basic introduction and time and space complexity analysis **[4 Hours]**

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.Cache optimization,Adversary lower bounds (for the comparison-based sorting algorithms). Using Recurrence relations and Mathematical Induction to get asymptotic bounds on time complexity. Master's theorem and applications.

Unit-II Divide and Conquer [4 Hours]

General strategy, Application of divide and conquer for solution of some computational problems like:Quick sort, Merge sort, Finding a majority element, Order statistics (randomized and deterministic algorithms), Efficient algorithms for Integer arithmetic (Euclid's algorithm, Karatsuba's algorithm for integer multiplication, fast exponentiation).

Unit-III Dynamic Programming [6 Hours]

General strategy, Application of dynamic programming for solution of some computational problems like:computing Fibonacci numbers, binomial coefficients, Matrix Chain multiplication,Coin change problem, 0-1 Knapsack, Traveling Salesperson Problem, Optimal Binary Search Tree construction, Shortest paths in directed acyclic graphs, All pair shortest path algorithm, Longest increasing subsequence problem, Largest independent set for trees.

Section2:Topics/Contents

Unit-IV Greedy and Backtracking strategy [4 Hours]

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

Backtracking: General strategy, backtracking strategy for some problems like:n-queen problem, graph coloring, subset sum problem, vertex cover, independent set, solution of SUDOKU puzzle.

(Note: for all the strategies discussed in Units II, III, and IV the student is expected to understand the essence of the strategies and not just the specific applications and should be able to apply suitable strategies for fresh problems)

Unit-V Introduction to complexity classes and NP-completeness [6 Hours]

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 (ILP)**).** Brief introduction to Linear Programming and modeling NP-complete problems using ILP.

Unit-VI Introduction to Randomized and Approximation algorithms [4 Hours]

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, Karger's Min-cut algorithm, coupon collector problem,

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.Introduction to quantum computation.

List of Tutorials (12):

1. Problem solving based on asymptotic notations, solution of recurrences.
2. Proving correctness of algorithms: some techniques
3. Problem solving based on Divide and Conquer strategy (Binary search interesting applications, counting inversions)
4. Advanced problem solving based on Divide and Conquer strategy (Discrete Ham-Sandwich theorem, efficient algorithm for Josephus problem)
5. Problem solving based on Dynamic Programming strategy (Largest sum contiguous block and generalizations, Optimal binary search tree (OBST) construction)
6. Advanced problem solving based on Dynamic Programming strategy (Winning strategy for two player games, Variants of shortest path algorithms)
7. Problem solving based on Greedy strategy with emphasis on proof of correctness.
8. Problem solving based on Backtracking strategy.
9. reducing NP problems to Integer Linear Programming.
10. Problem solving based on complexity classes, NP-completeness.
11. Problem solving based on Randomized Algorithms
12. Problem solving based on Approximation Algorithms

List of Practical's (Minimum Six):

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

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 Home AssignmentsTopics:

List of Design Based Home Assignments:

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

List of Case Study Based Home Assignments:

1. AKS primality test

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

List ofBlog Based Home Assignment:

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

List of Survey Based Home Assignments:

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

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.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Presentation: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (Written): 60 Marks converted to 30 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

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*

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: 978-3642084690Gerd Keiser, MC Graw Hill International edition, optical fiber communication , third edition*

MOOCs Links and additional reading material:

1. www.nptelvideos.in

Course Outcomes:

The student will be able to –

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:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO2	PSO3	PSO4
1	2	3	2	1			2					2	3		2	
2	2	3	3	1			2					2	3		2	
3	2	3	2	1			2					2	3		2	
4	2	3	3	1			2					2	3		2	
5	2	3	3	1			2					2	3		2	
6	2	3	3	1			2					2	3		2	
Avg	**2**	**3**	**2.66**	**1.0**			**2.0**					**2.0**	**3**		**2**	

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65 L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

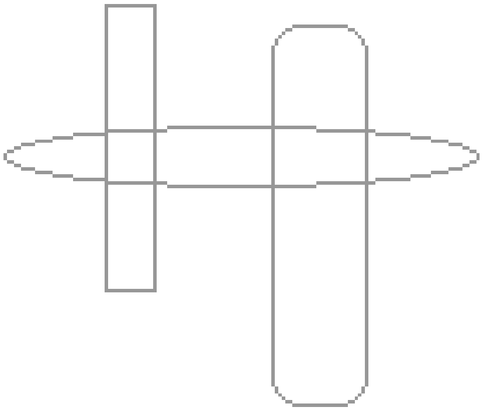
CO1 – L1, CO2 – L3, CO3 – L2, CO4 – L3, CO5 – L4 and CO6 – L5

Future Course Mapping:

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.



CI3003: Artificial Neural Networks

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites: Mathematics for AI and Artificial Intelligence

Learning Objectives:

1. To learn how to design AI based learning system and their evaluation methods
2. To learn ANN learning types and ANN training process
3. To learn Single Layer Perceptron Classifiers
4. To learn Multilayer Feedforward Networks
5. To learn Competitive learning Neural Network
6. To learn Optimization of Neural Networks

Course Relevance:

Artificial Neural Networks are at the forefront of the AI revolution, with applications spanning across multiple domains, driving technological advancements, and creating significant career opportunities. Understanding ANNs is crucial for anyone looking to engage with modern AI technologies, contribute to innovative research, or develop cutting-edge solutions in various fields.

Section 1: Topics/Contents

Unit-I: Learning System and evaluation parameters [4 Hours]

Well posed learning problem, Designing a learning system, Issues in AI 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.

Unit II: Introduction to Artificial Neural Networks (ANNs) [5 Hours]

Neural Computation, History of Artificial Neural Systems Development, Biological Neurons and Their Artificial Models, Models of Artificial Neural Networks, Neural Network Learning Rules, Perceptron Learning Algorithms. Supervised learning, unsupervised learning, and reinforcement learning, Training Process: Overview of the neural network training process and the role of training data., Initialization of Weights: Importance of initializing weights and common strategies (e.g., random, Xavier, He initialization). Hebbian Learning, Competitive, Boltzmann Learning,

Unit III: Single Layer Perceptron Classifiers [5 Hours]

Classification Model, Features, and Decision Regions, Discriminant Functions, Linear Machine and Minimum Distance Classification, Nonparametric Training Concept, Training and Classification Using the Discrete Perceptron: Algorithm and Example, Single-Layer Continuous Perceptron Networks for Linearly Separable Classifications, Multicategory Single-Layer Perceptron Network

Section2: Topics/Contents

Unit IV: Multilayer Feedforward Networks [5 Hours]

Linearly Nonseparable Pattern Classification, Delta Learning Rule for Multi-Perceptron Layer, Generalized Delta Learning Rule, Feedforward Recall and Error Back-Propagation Training, Learning Factors, Classifying and Expert Layered Network, Multilayer Networks, Backpropagation algorithm, case study to implement MLP.

UNIT V Competitive learning Neural Network [5 Hours]

Components of CL network, Pattern clustering and feature mapping network, ART networks, Features of ART models, character recognition using ART network. Self-Organization Maps, (SOM): Two Basic Feature Mapping Models, Self-Organization Map, SOM Algorithm, Properties of Feature Map, Computer Simulations, Learning Vector Quantization, Adaptive Pattern Classification. Application and analysis of ART1 &amp; ART2, Case study

Unit VI: Optimization of Neural Networks [4 Hours]

Data Preparation and Preprocessing, Weight Initialization Techniques, Loss Functions (Mean Squared Error (MSE) and Cross-Entropy Loss.) and Backpropagation, Learning Rate strategies and its Importance, Gradient Descent Variants (SGD, Adam, RMSprop), Overfitting and Regularization Techniques (L1, L2, Dropout), Hyper-parameter Tuning, Handling Vanishing and Exploding Gradients

List of Tutorials (14):

12. Examples on Bias and Variance
13. Examples on Overfitting and Inductive bias
14. Examples on Precision and recall
15. Examples on ANN learning
16. Examples on ANN learning
17. Examples on ANN learning

18. Examples on Single Layer Perceptron Classifiers
19. Examples on Single Layer Perceptron Classifiers
20. Examples on Multilayer Layer Perceptron Classifiers
21. Examples on Multilayer Layer Perceptron Classifiers
22. Examples on Competitive learning Neural Network
23. Examples on Competitive learning Neural Network
24. Examples on ANN optimization
25. Examples on ANN optimization

List of Practical's (Minimum Six):

1. Write a program to visualize popular activation functions used in neural networks, aiding in understanding their behaviour and suitability for different tasks.
2. Write a program to use a Perceptron neural network to recognize even and odd numbers represented in ASCII form, illustrating the basic principles of a single-layer neural network.
3. Write a program to demonstrate the perceptron learning law with decision regions, visually showcasing how a perceptron learns and separates different classes.
4. Write a program to implement the training process of an Artificial Neural Network, covering forward propagation and backpropagation, fundamental steps in neural network training.
5. Write a program to create a neural network architecture from scratch, focusing on multi-class classification with customizable parameters such as hidden layers, neurons, non-linearity, and optimization algorithm.
6. Write a program to illustrate an ART (Adaptive Resonance Theory) neural network, showcasing its self-organizing and adaptive capabilities.
7. Write a program to implement of AND/NAND gate using feed forward neural network.
8. Write a program to implement of OR/ NOR gate using feed forward neural network.
9. Write a program to implement of Ex-OR gate using feed forward neural network.

List of Course Project areas:

1. Security: Use of ANN in Cryptographic Applications.
2. Banking: Use of ANN for Credit Scoring System
3. Music: Use of ANN for Automatic Music Generation
4. Autonomous Vehicle: Use of ANN for Vision and Control in Autonomous Flying Vehicle.
5. GPS: USE of ANN in Global Positioning System.
6. Recognition: Use of ANN for Handwritten Recognition.
7. Stock Market: Use of ANN for stock Market prediction.
8. Training System: Use of ANN for Web based Training Systems
9. Vehicle Security: Use of ANN to build Vehicle Security System

List of Topics for PPT/Seminar:

1. ANN for Cryptographic Applications.
2. ANN for Credit Scoring System
3. ANN for Automatic Music Generation

4. ANN for Vision and Control in Autonomous Flying Vehicle.
5. ANN in Global Positioning System.
6. ANN for Handwritten Recognition.
7. ANN for stock Market prediction.
8. ANN for Web based Training Systems
9. ANN to build Vehicle Security System

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.

Assessment scheme covers following aspects of Modified Blooms Taxonomy:

L2 Understanding, L3 Apply, L3 Design, L3 Apply, L4 Analyze and L5 Evaluate
Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Seminar/Presentation: End Semester Examination: 100 Marks converted to 20 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (Written): 60 Marks converted to 30 Marks

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

1. Introduction to Artificial Neural Systems, JACEK M. ZURADA, Jaico Publishing House; First Edition, 1994
2. Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis & Applications, S.Rajasekaran, G. A. Vijayalakshami, PHI, 2007
3. Neural Networks and Learning Machines by Simon S. Haykin, Pearson; 3rd edition 12 March 2009

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

1. Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems by GeronAurelien
2. Foundations of Neural Networks, Fuzzy Systems, and Knowldge Engineering, Nikola K. Kasabov, MIT Press, 1998
3.https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939

MOOCs Links and additional reading material:

1. NPTEL Course “Neural Networks and Application”, Prof. SomnathSengupta, IIT Kharagpur Link of the Course:https://nptel.ac.in/courses/117105084

Course Outcomes:

The student will be able to –

1. Decide on learning system and their evaluation parameters
2. Select appropriate ANN learning type and training process
3. Propose suitable Single Layer Perceptron Classifier
4. Propose suitable Multilayer Feedforward Networks

5. Apply Competitive learning Neural Network
6. Design, Implement, Evaluate and Optimize ANN based applications

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO 3	PO4	PO5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	
1	3	**2**	2	2	**-**	2	-	-	1	-	-	1	1	2	1	
2	3	3	**3**	**3**	3	2	2	2	1	-	-	2	2	3	2	
3	3	3	2	2	3	2	-	**2**	1	-	-	2	2	3	2	
4	3	3	2	2	3	**2**	-	2	1	-	-	2	2	3	2	
5	3	3	3	3	3	2	-	2	**1**	-	**-**	**3**	**3**	3	**3**	
6	3	3	3	3	3	3	-	3	3	2	2.0	3	3	3	3	
Avg	**3**	**2.83**	**2.5**	**2.5**	**3**	**2.16**	**2**	**2.2**	**1.33**	**2.0**	**2.0**	**2.16**	**2.16**	**2.83**	2.16	

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

CO Attainment levels:

Weights for attainment levels: L1 - Easy-0.75 L2 - Comfortatble-0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L2, CO2 – L3, CO3 – L4, CO4 – L4, CO5 – L4 and CO6 – L5

Future Course Mapping:

Advanced course on Deep learning including Auto encoders and Boltzmann machines, Reinforcement Learning etc

Job Mapping:

AI Scientist, AI System Designer, AI Developer, AI Data Analyst

Syllabus Template **FF No. : 654**

CI3004: Cloud Computing

Credits: 4....... **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites: Operating Systems, Computer Networks, Database Management System

Course Objectives:

1. To become familiar with cloud computing and its ecosystem
2. To acquire basics of virtualization and its importance
3. To evaluate in-depth analysis of Cloud Computing capabilities and its services.
4. To configure and implement storage services.
5. To analyze different cloud-based services to meet a set of given requirements.
6. To design security aspects for cloud computing

Course Relevance: Cloud computing to enable transformation, business development and agility in an organization.

SECTION-I Topics and Contents:

Unit-I Introduction to Cloud Computing [4 Hrs]

Recent trends in computing, Cluster computing, Distributed computing, Evolution of cloud computing, Cloud versus traditional architecture, Cloud Computing Architecture, Google Cloud architecture, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), Public cloud, Private cloud, Hybrid cloud, Community cloud

Unit-II Virtualization [6 Hrs]

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, Basics of virtualization and implementation challenges. System virtualization technologies-architectures and internals. KVM, Xen, VMware. Amazon Elastic Compute Cloud EC2 as computing service.

Memory virtualization-virtualization techniques, ballooning, deduplication and sharing. Network and storage virtualization, Virtual machine migration and replication techniques pre-copy and post-copy techniques, applicability to system availability.

Unit-III Cloud Services [4 Hrs]

Service Oriented Architecture (SOA), Web services, Web 2.0, Web OS. Introduction to IaaS, PaaS, SaaS. 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) Docker flow, orchestration with Docker, dynamic linking and legacy linking of containers. The GCP Console, understanding projects, Billing in GCP, Install and configure Cloud SDK, Use Cloud Shell, GCP APIs.

SECTION-II Topics and Contents:

Unit-IV Cloud Storage [4 Hrs]

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. OpenStack: NOVA, Neutron, Keystone Cinder, Swift and Glances, VMware Suit, Apache Cloud Stack, Data Lakes, Snowflake.

Unit-V Service Management [4 Hrs]

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., Introduction to configuration and management tools Ansible, Architecture of DevOps.

Unit-VI Cloud Network and Security [6 Hrs]

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. Introduction to security in the cloud, the shared security model, Encryption options

List of Tutorials (Any Thirteen)

List of Tutorials:

Unit-I Introduction to Cloud Computing

1) Install VirtualBox/VMware Workstation with different Linux or Windows Operating Systems.
2) Study Google Cloud Architecture.

Unit-II Virtualization

3) Find a procedure to launch virtual machine
4) Find a procedure to transfer the files from one virtual machine to another virtual machine.

Unit-III Cloud Services

5) Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

6) Install Google App Engine. Create hello world app and other simple web applications using python/java.

Unit-IV Cloud Storage

7) Launch the Web Applications using GAE launcher.

8) Install Hadoop single node cluster and run simple applications like wordcount.

Unit-V Service Management

9) Use AWS Pricing Calculator: Create estimate for EC2 Compute cost for VM instance. Use region closest to you. Find On demand cost and compare the pricing for other regions.

Unit-VI Cloud Network and Security

9) Launch EC2instane and explore Public/Private/Elastic IP

Practical's:

List of Practical's (Any Six)

Unit-I Introduction to Cloud Computing

1) To setup AWS accounts and launch instances.

Unit-II Virtualization

2) To install an OS using VirtualBox/ VMWare Workstation. Add Storage to create new virtual disk.

3) To Deploy Virtual Machine on hypervisor such as KVM, ESXi. Take Backup and Migrate them.

Unit-III Cloud Services

4) To use Infrastructure as a Service to facilitates for creating and deleting compute resources. Create network and attach volumes to run instances.

5) To install docker on window/linux and build docker image from docker hub.

6) Deploy a stateless/stateful application on Kubernetes cluster.

Unit-IV Cloud Storage

7) To work on different Cloud Storage Services.

Unit-V Service Management

8)To create login into AWS and use S3 Bucket Service for storage.

Unit-VI Cloud Network and Security

9) Develop elastic services for dynamic load scenario using AWS APIs. Build load balancer and explore on scalability, fault detection and performance.

Course Projects:

List of Course Project Topics

1. Creating Google Account to store files and programs.
2. Creating Account to Store Images.
3. Creating a Warehouse Application in SalesForce.com

4. Creating an Application in SalesForce.com using Apex programming Language.
5. To study and implement Web services in SOAP for JAVA Applications.

6. Implementation of Para-Virtualization using VMWare 's Workstation/ Oracle's Virtual Box and Guest Operator System.
7. Installation and Configuration of Hadoop.
8. AWS Case Study: Amazon.com.
9. Case Study of Google App Engine.
10. Case Study of Face book.

Seminars:

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
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

Group Discussion:

List of 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:

List of Design Based Home Assignments

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).

List of Case Study Based Home Assignments

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.

List of Blog Based Home Assignment

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.

List of Survey Based Home Assignments

1. Disaster Recovery in Cloud Computing.
2. Cloud Economics.
3. Data archiving solutions.
4. Salesforce.
5. Dropbox.

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.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Presentation: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (MCQ): 60 Marks converted to 30 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

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:

On the completion of course, student will able to

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:

CO	Program Outcomes (PO)											PSO			
	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	
CS3226.1	2	1			1								2	2	3	
CS3226.2	2	2	1	1	1								2	2	3	
CS3226.3	3	2	2	2	2		3	3					2	2	3	
CS3226.4	3	2	2	2	3	3			3				2	2	3	
CS3226.5	3	3	1	3	3				1		2		2	2	3	
CS3226.6	2	2	1	3	1					3		3	2	2	2	
Average	2.50	2.00	1.40	2.20	1.83	3.00	3.00	3.00	2.00	3.00	2.00	3.00	2.0	2.0	2.83	

CO attainment levels:

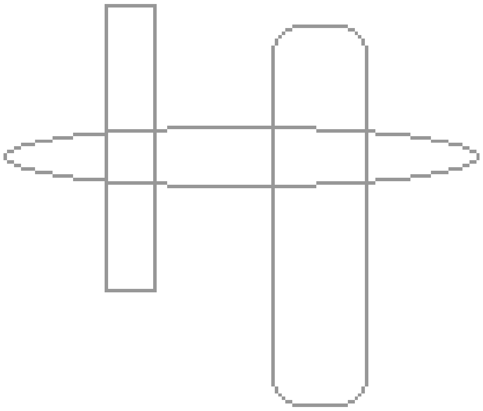
Attainment Levels:1,2,3,5,4,3

Future Course 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:

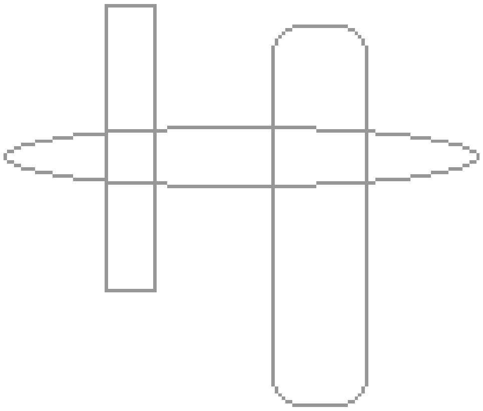
Cloud Architect, Cloud Engineer, Cloud Administrator, Solutions Architect - Cloud Computing - AWS / Kubernetes, Cloud Computing Technical Consultant, Associate Cloud Computing Engineer, Cloud Computing Trainer



T. Y. B. Tech. Computer Science & Engineering (Artificial Intelligence)

AY 2024-25

Module VI Course Content



Syllabus Template **FF No. : 654**

CI3007: Software Engineering

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites:

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 differentiate feasible and competing system requirements, indicating correct real world problem scope and preparing stepwise system conceptual model,
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 using UML diagrams,
5. To design a system architecture and map it with a suitable architectural style,
6. To comprehend the nature of design patterns and apply these patterns in system design.

Course Relevance:

Given that Software Engineering is built upon the foundations of Computer Science as well as Computer Engineering, a Software Engineering curriculum can be focused on two perspectives - a Computer Science-first or Software Engineering-first perspective. Software engineering spans the entire software lifecycle. It involves creating high-quality and reliable programs in a systematic, controlled, and efficient manner using formal methods for specification, analysis, design and evaluation of proposed systems. It requires suitable software development techniques and processes that successfully scale to large applications, which should satisfy timing, size, and security requirements all within acceptable application/project budgets and deadlines. For these reasons, Software Engineering requires both the analytical and descriptive tools and techniques developed in Computer Science and the rigor that the Computer Engineering discipline brings to the reliability and trustworthiness of the systems that software developers design and implement, while working as a cohesive team.

<u>Section 1: Topics / Contents</u>

Unit-I Software Engineering Paradigms: **05 Hours**

Process Models: Code-and-Fix Model, Waterfall Model, Rapid Application Development Model, Incremental Model, Evolutionary Model and Others.

Unit-II Requirements Engineering: **05 Hours**

Requirements Engineering Tasks, Requirement Elicitation Techniques, Functional, Non- Functional and Domain Requirements, Requirements Characteristics, Eliminating Requirement Ambiguities, Conflict Identification and Resolution, Requirement Qualities, Requirement Specification, System Scope Determination and Feasibility Study.

Unit-III Agile Methodology: **04 Hours**

Landscape of Agile and Planned Methods, Definition - Scrum, Scrum Origins, Scrum Framework, Agile Principles, Sprints, Requirements, User Stories, Product Backlog, Roles: Product Owner, Scrum Master, Development Team, Managers, Scrum Team Structures, Scrum Planning.

Section 2: Topics/Contents

Unit-IV Static and Dynamic Interaction Modeling: **05 Hours**

Static Behavior: Use Case, Use Case Diagram, Class Diagram, Component Diagram, Deployment Diagram, Dynamic Behavior: Sequence Diagram, Collaboration Diagram, Activity Diagram, Communication Diagram, Interaction Diagrams.

Unit-V Software Architecture Design: **05 Hours**

Design Model, Design Qualities, Characteristics of Design Activities, Design Principles, Cohesion and Coupling, Software Architecture Vs Software Design, Software Reuse, Design Heuristics, Layered Architecture, Client-Server Architecture, Pipe-Filter Architecture, Model-View Controller Architecture.

Unit-VI Design Patterns: **04 Hours**

Definition, Describing Design Pattern,
Creational Patterns: Abstract Factory, Builder, Factory Method, Prototype
Structural Patterns: Adapter, Bridge, Composite, Decorator, Façade,
Behavioral Patterns: Chain of Responsibility, Command, Interpreter.

List of Tutorials:

1. Requirement Engineering,
2. System Requirement Specification,
3. Scrum Artifacts,
4. User Stories and Use Cases,
5. Product Backlog Development,
6. Burn-up and Burn-down Chart Development and Management,
7. Software System Analysis and Design: UML Static Diagram,
8. Software System Analysis and Design: UML Dynamic Diagram,
9. Software Architecture Design,
10. Use of Design Patterns,
11. Software Testing,

12. Automated Testing,
13. Project Management Techniques.

List of Practicals (Minimum SIX):

1. To prepare a Statement Of Work (SOW) document, which addresses the vision, goals and objectives of the real-world problem.
2. To prepare a Software Requirement Specification (SRS) document, based on several types of system requirements, such as functional and non-functional requirements.
3. To document a product backlog for the project aimed at maintaining a prioritized queue of project requirements.
4. To develop a Sprint-plan and Sprint-design indicating detailed activity planner accommodating user story points.
5. To prepare Class Collaboration-Responsibility (CRC) cards for the Conceptual classes traced from the system analysis phase.
6. To develop a static structure of the target system with a Class Diagram using all components of it.
7. To decompose and organize the problem domain area into broad subject areas and identify the use cases to show them in a Use Case Diagram.
8. 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.
9. To depict the dynamic behavior using a detailed Activity Diagram.
10. To prepare an Architecture Diagram with appropriate design patterns. Suitable Architectural Styles shall be selected and the structural elements shall be well-documented.

List of Course Projects:

1. Automated Parking Lot Identifier,
2. Healthcare Software,
3. Financial Application,
4. Appraisal System,
5. Smart Project Administrative System,
6. Translator for Agriculture System,
7. Development of Applications using Agile Methodology,
8. Development of SMART Mobile Applications,
9. Graphics-based Password Identification System
10. System Security Application

List of Course Seminar Topics:

1. Mobile Apps and App Store Analysis,
2. Automated Reasoning Techniques,
3. Autonomic and Self-Adaptive System,
4. Component-based Software Engineering,

5. Computer-Supported Cooperative Work (CSCW),
6. Configuration Management and Deployment,
7. Crowd-Sourced Software Engineering,
8. Cyber-Physical System,
9. Data-driven Software Engineering,
10. Dependability, Safety and Reliability.

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

Case Study:

1. Software Economics and Metrics
2. Machine Learning in Software Engineering
3. Software Evolution and Maintenance
4. Software Modeling and Design
5. Software Product Lines

Blog

1. Mining Software Engineering Repositories
2. Model-driven Engineering
3. Parallel, Distributed and Concurrent systems
4. Recommendation Systems
5. Refactoring

Surveys

1. Reverse Engineering
2. Safety-Critical Systems
3. Security, Privacy and Trust
4. Software Architecture
5. Software Reuse
6. Software Testing

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Group Discussion: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (MCQ): 60 Marks converted to 30 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

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

1. *Ian Sommerville, 'Software Engineering',Pearson, 10th Edition, 2017, ISBN-13: 978-9332582699.*
2. *Kenneth Rubin, 'Essential SCRUM: A Practical Guide To The Most Popular Agile Process', Addison-Wesley, 2012, ISBN-13: 978-0-13-704329-3.*
3. *Tom Pender, 'UML Bible', John Wiley & Sons, 2003, ISBN - 0764526049*

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

1. *SorenLauesen, 'Software Requirements: Styles and Techniques, Addison Wesley, 2002, ISBN 0201745704.*
2. *Dean Leffingwell, 'Agile Software Requirements', Addison-Wesley, 2011, ISBN-13: 978-0-321-63584-6.*
3. *Grady Booch, James Rambaugh, Ivar Jacobson, 'Unified Modeling Language User's Guide', 2nd Edition, Addison-Wesley 2005, ISBN – 0321267974.*
4. *Erich Gamma, Richard Helm, Ralph Johnson, 'Design Patterns: Elements of Reusable Object-Oriented Software', Addison-Wesley Professional, 1994, ISBN-13: 978-0201633610.*
5. *Paul Clements, Felix Bachmann, Len Bass, David Garlan, 'Documenting Software Architectures: Views and Beyond', Addison-Wesley Professional, 2003, ISBN-13: 9780201703726.*

MOOCs Links and additional reading material:

www.nptelvideos.in, www.coursera.com, www.udemy.com

Course Outcomes:

The student will be able to –

1. Compare Software Development Process Models and justify process maturity through application of Software Engineering principles and practices,
2. Differentiate competing and feasible system requirements identifying problem scope in the real-world,
3. Apply agile principles and practices through scrum framework,
4. Design UML diagrams through efficient system analysis, using identified design specifications
5. Formulate system architecture as per a suitable architectural style,
6. Apply relevant design patterns for effective system design.

CO-PO Map:

	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	
1	2	3	2									3	2	3		
2	2	3	2									3	2	3		
3	3	2	3		3	2	2	2			2	3	2	3		
4	3	2	3		3	2	2	2	2		2	3	2	3		
5	3	2	3		3	2	2	2			2	3	2	3		
6	3	2	3		3	2	2	2		3	2	3	2	3		
Average	2.66	2.33	2.66		3	2	2	2	2.0	3.0	2.0	3.0	2.0	3.0		

CO Attainment levels:

Weights for attainment levels: L1 - Easy- 0.75 L2 - Comfortable - 0.7 L3 – Medium – 0.65
L4 – Somewhat difficult – 0.6 L5 – Difficult – 0.55

CO1 – L1, CO2 – L2, CO3 – L3, CO4 – L4, CO5 – L4 and CO6 – L5

Future Course Mapping:

Software testing and Quality Assurance, Service-Oriented Software

Job Mapping:

Application Architect, Project Designer, SCRUM Role Player, Project Manager

Syllabus Template **FF No. : 654**

CI3008: Cyber Security and Blockchain

Credits: 4....... **Teaching Scheme Theory: 2 Hours/Week**

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites: Computer Networks

Course Objectives:

1. Apply cryptographic techniques and security protocols to secure systems and networks.
2. Identify, resolve, and mitigate programming bugs and cyber threats.
3. Design secure systems using blockchain technology and ensure application security.
4. Understand and apply cloud security and physical security principles.
5. Integrate AI in cyber security and develop business continuity and disaster recovery plans.
6. Implement ethical hacking practices and perform effective penetration testing.

Course Relevance:
Cyber Security teaches how to protect operating systems, networks, and data from cyber attacks, monitor systems, and mitigate threats, aiming to develop skills to prevent attacks and protect data privacy.

Section 1: Topics/Contents

Information security

- **Key Security Properties:** Confidentiality, Integrity, Availability.
- **Risk Management:** Understanding governance policies, frameworks, laws, regulations, guidelines, and compliance.
- **Symmetric Key Cryptography:** Role of random numbers and nonce in security, importance of prime numbers, GCD, Euclid's Algorithm, Extended Euclid's algorithm.
- **Data Encryption Standard (DES):** Block cipher, stream cipher, Feistel structure, block cipher modes, S-DES, attacks on DES, S-AES, AES.

- **Public Key Cryptography:** RSA algorithm, key generation, attacks on RSA.
- **Elliptic Curve Cryptography (ECC):** Elliptic curves over real numbers and Zp, elliptic curve arithmetic.

Network Security

- **Certificates and Hashing:** Properties of hash functions, HASH + SALT, hashing algorithms (SHA1, SHA2).
- **Authentication and Authorization:** Network access control (SHA-512, Kerberos, and multifactor authentication).

- **Transport-Level Security:** Web security considerations, Secure Sockets Layer (SSL), Transport Layer Security (TLS), HTTPS standard, Secure Shell (SSH) application, IPSec.
- **Application Security:** Security by design, writing secure code, static and dynamic application security testing (SAST and DAST), interactive application security testing (IAST), Integrated Security in DevOps, OWASP, Application Security Services,

Section 2: Topics/Contents

Cyber Attacks and Penetration Testing (06 Hours)

- **Cyber Ethics:** Threats, threat modeling, injections, sniffing, and types of attacks.
- **Security Vulnerabilities:** risk, attack types, countermeasures.
- **Protocol Vulnerabilities:** DoS and DDoS, session hijacking, ARP spoofing.
- **Software Vulnerabilities:** Phishing, buffer overflow, cross-site scripting attack, ransomware, SYN-flooding, SQL-injection, DNS poisoning.
- **Penetration Testing:** Difference from automated vulnerability scans, objectives and limitations of a pen test, scoping and planning pen tests, executing pen tests and managing findings. Introduction to SDL (Secure Development Lifecycle) – Merging Security into SDLC,

Physical Security and Forensics (04 Hours)

- **Physical Security:** Physical access types, crime prevention through environmental design (CPTED).
- **IoT Security:** Definitions of OT, IoT, IIoT, and ICS, most widely used protocols in IoT environments (MQTT and CoAP).
- **Business Continuity (BC):** RTP/RPO, RTO, MTPD, ISO 22301 standard for business continuity management, importance, differences between BCMS and DRMS, risk management, testing, maintenance., Operation Resilience,
- **Digital Forensics:** Introduction to digital forensics, data recovery, OS forensics, email crimes and violations, cyber forensics.

Cloud Security

Principles / Key Concepts of Cloud Security: Overview of cloud security principles and key concepts.

- **Threats and Risks in Cloud Security:** Diverse types of threats and risks associated with cloud security.
- **Importance of Security Measures in Cloud Security:** Importance of implementing security measures in cloud environments.
- **Solutions for Cloud Security:** Effective solutions to address cloud security challenges.

Role of AI in Cyber Security: Examination of how AI is integrated into cyber security.

- **Challenges and Opportunities of AI in Cyber Security:** Analysis of the challenges and opportunities presented by AI in the field of cyber security.

Blockchain

- **Decentralized Systems & Distributed Ledger Technology:** Blockchain computing power, hash, and Merkle tree with hands-on examples.
- **Use-Cases of Blockchain:** Different types of blockchain including public and private blockchain, consensus and types of consensus with examples.
- **Smart Contracts:** Need for smart contracts, developing smart contracts, programming basics of Solidity (data types) and advanced Solidity, EVM in relation to smart contracts and gas price, running and debugging smart contracts in Remix, deploying and debugging smart contracts with Truffle.

List of Tutorials (13)

1.Mathematical background for cryptography: modulo arithmetic, GCD (Euclid's algorithm), algebraic structures (Groups, Rings, Fields, Polynomial Field).

2.Chinese remainder theorem.

3. Diffie-Hellman key exchange: Algorithm, Key exchange protocol, Attack.

4. ECC over Diffie-Hellman key exchange.

5. Study of certificates and hashing algorithms.

6. Network access control and transport-level security.

7. Security by design and writing secure code.

8. Static and dynamic application security testing.

9. Study of Snort.

10. Nessus: a Security Vulnerability scanning tool.

11. Metasploit/Ollydbg.

12. Testing for Brute Force Password.

13. Testing for SQL Injection.

14. Computer forensics, Facebook forensic, mobile forensic, cyber forensic, digital forensic.

15. Source Code Analysis Tools.

16. OWASP Zed Attack Proxy (ZAP).

17. Study of various types of Blockchain, Connecting the Metamask wallet with the local Ganache network.

18. Simulation of Blockchain.

19. Creating Smart Contract using Solidity and Remix IDE.

20. Study of DOA and DAPP.

List of Practicals (Minimum Six)

Section-I:

- Simplified DES implementation.
- Simplified AES implementation.

- Encryption and Decryption by RSA algorithm.
- Implementation of ECC over Diffie Hellman Key Exchange Protocol.

- Implementation of authentication algorithms.
- Implementation of SHA.

Section-II:

- Acquisition of System Information/ RAM/Volume Shadow Copy/Detecting Encryption in information.
- Vulnerabilities finding in Mobile/ computer/ digital devices.

- Forensic of Disc Image/ Registry/ Meta data/ RAM.
- Digital forensic of images.
- Forensics of Video alteration.

- Implement and demonstrate the use of the following in Solidity: Variable, Operators, Loops, Decision Making, Strings, Arrays, Enums, Structs.
- Implement and demonstrate the use of the following in Solidity: Functions, Function Modifiers, View functions, Pure Functions, Mathematical functions, Cryptographic functions.
- Use Geth to configure a private Blockchain node in our machine.
- Cryptography in Blockchain, Merkle root tree hash.
- Creating Transactions using Solidity and Remix IDE.
- Case Study on Hyperledger Fabric.

List of Course Project areas:

Course Project 01 Statement: 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.

Course Project 02 Statement: Securing Video Conferencing App for online meetings

Course Project 03 Statement: Steganography for Image/Video/Files

Course Project 04 Statement: Secure Image display on online social media.

Course Project 05 Statement: Secure transfer of government subsidies to farmers/BPL people/ students etc

Course Project 06 Statement: Authentication of users for various applications for integrity, availability, confidentiality.

Course Project 07 Statement: Implementing a system for detecting the modification of videos/images on social media

Course Project 08 Statement: Secure App for online exams detecting Keystroke and camera movements.

Course Project 09 Statement: A system to detect the difference between the voice edited in the audio/video

Course Project 10 Statement: A System to check the vulnerabilities in the websites.

Course Project 11 Statement: Decentralized (Uber)Peer to Peer Carpooling

Course Project 12 Statement: Decentralized Skill Verification System

Course Project 13 Statement: Decentralized talent acquisition (like Nokari.com)

Course Project 14 Statement: Decentralized gaming DAPP(earn coin through game)

List of Course Seminar Topics

Seminar 01 Statement: Blockchain architecture and its implementation

Seminar 02 Statement: Cloud Security

Seminar 03 Statement: Mobile Security

Seminar 04 Statement: IoT and Security Issues/ Security Models for IoT

Seminar 05 Statement: Dark web

Seminar 06 Statement: Docker Security

Seminar 07 Statement: Access control methods for online social media and various organizations

Seminar 08 Statement: Security of Android Vs IOS

Seminar 09 Statement: Machine learning and SCADA Security

Seminar 10 Statement: Security Applications for Smart

List of Design Based Home Assignments

HA_D 01 Statement: Design a secure system using cryptography techniques for security of multimedia files.

HA_D 02 Statement: Design a secure system using steganography for hiding data files in image/video

HA_D 03 Statement: 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.

HA_D 04 Statement: Design a secure system using SSL/TLS/IPSec for the various organizations

HA_D 05 Statement: Design a system for the analysis of cyber crime using various cyber forensic techniques and compare each technique with respect to integrity, confidentiality, availability

List of Case Study Based Home Assignments

HA_CS 01 Statement: How to improve the security of social media? Write a detail case study

HA_CS 02 Statement: 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.

HA_CS 03 Statement: Write a detail case study about the banking security flows and solutions to these flows.

HA_CS 04 Statement: Give a detail case study of the antivirus system giving the flows and solutions to it.

HA_CS 05 Statement: Perform the detail case study of various operating systems used for mobile devices and give a secure solution to one for widely used OS.

List of Blog Based Home Assignment

HA_Blog 01 Statement: Dark Web

HA_Blog 02 Statement: Crypto currency and Economy

HA_Blog 03Statement:.Cybercrime and solutions

HA_Blog 04 Statement: Authentication and Access control for social media

HA_Blog 05 Statement: Cyber forensic and Cyber laws

List of Survey Based Home Assignments

HA_Survey 01 Statement: Survey on various blockchain related issues/ cryptocurrency/ application systems developed using blockchain

HA_Survey 02 Statement: Survey on various authentication and access control methods for different applications

HA_Survey 03 Statement: Steganography and Biometric Systems for authentication

HA_Survey 04 Statement: Survey of various attacks and its effect on Indian economy and its analysis

HA_Survey 05 Statement:Problems over Integer Lattices: A Study

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.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Presentation: End Semester Examination: 100 Marks converted to 20 Marks
Theory: End Semester Examination (Written): 60 Marks converted to 30 Marks

Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

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*
4. *AmbadasTulajadasChoudhari, Arshad SarfarzAriff, Sham M R, "Blockchain for Enterprise Application Developers" Willey publications, ISBN: 9788126599967,2020*

Hyperledger Fabric - https://www.hyperledger.org/projects/fabric

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.*
6. *Mastering Blockchain: Deeper insights into decentralization, cryptography, Bitcoin, and popular Blockchain frameworks by Bashir, Imran,2017.*

Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016

MOOCs Links and additional reading material:

1. Cryptography And Network Security By Prof. Sourav Mukhopadhyay, IIT Kharagpur Cryptography and Network Security - Course (nptel.ac.in)
2. Information Security and Cyber ForensicsBy Prof. Pratosh Bansal Devi Ahilya Vishwavidyalaya, Indore, Information Security and Cyber Forensics - Course (swayam2.ac.in)
3. Blockchain and its Applications By Prof. Sandip Chakraborty, Prof. Shamik Sural IIT Kharagpur Blockchain and its Applications - Course (nptel.ac.in)

Course Outcomes:

The student will be able to –

1. Demonstrate cryptographic techniques using a mathematical approach by examining nature of attack.

2. 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
3. Justify various methods of authentication and access control for application of technologies to various sections of industry and society.
4. Identify and establish different attacks on the system.
5. Estimate future needs of security for a system by researching current environment on a continuous basis for the benefit of society.
6. Analyze the need of Decentralized system and implement using blockchain technology.

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
1	3	**3**		3	**2**	2		3	1				3		1	3
2	3	2	3	2			2	3	3	3	2	1				2
3	2	3	**3**		1	2		3	1						1	3
4	3	3	1	**3**	3	3	3	**3**	1			3		3	1	3
5	2	2	3	2	1		2		3	3	3	3				
6	3	2	1	1	3	3	3	3			**3**	**2**		3	3	3
Avg	**2.67**	**2.5**	**2.2**	**2.2**	**2**	**2.5**	**2.5**	**3**	**1.8**	**3**	**2.67**	**2.25**	**3**	**3**	**1.5**	**2.8**

CO Attainment levels:

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

Future Course Mapping:

Cloud Computing and Security, IoT Security, Ethical Hacking &Cyber Forensics

Job Mapping:

Security Engineer/Network Security Engineer

Information Security Analyst

Cyber Security Analyst

Cyber Security Associate

Manager-Information Security Services

Security Consultant

Penetration Testing Engineer

Syllabus Template **FF No. : 654**

CIS3009: Deep Learning

Credits: 4 **Teaching Scheme Theory: 2 Hours/Week**
Tutorial: 1 Hours/Week
Lab: 2 Hours/Week

Course Prerequisites:

Linear algebra, probability theory and statistics,Artificial Neural Network

Course Objectives:

1.To explore data preprocessing methods, including feature selection and dimensionality reduction techniques like PCA and LDA.

2. To Apply various supervised learning algorithms, including linear and kernel-based models.

3.To evaluate classification methods and their applications in solving problems.

4. To understand the importance of deep learning and is variants

5. To understand the basics of Recurrent neural network models of NN

6. To build deep nets with applications to solve real world problem

Course Relevance:

In today’s data-driven landscape, machine learning and deep learning are critical for addressing complex industry challenges and driving innovation. As businesses increasingly rely on data analytics for decision-making, machine learning offers essential tools for predictive modeling, customer insights, and operational efficiency. Deep learning, with its ability to analyze vast amounts of unstructured data, is revolutionizing sectors such as healthcare through improved diagnostics, finance through algorithmic trading, and autonomous systems via enhanced perception and decision-making capabilities. This course prepares students to meet the growing demand for expertise in these technologies, equipping them with the skills to develop intelligent solutions that can transform industries and improve overall productivity.

Section 1: Topics/Contents

Unit-I : Introduction Machine Learning **[4 Hours]**

Motivation and role of machine learning in computer science and problem solving,Machine Learning Workflow,Introduce paradigms of Learning,Data Preprocessing and Feature Engineering,Feature Selection and Extraction Techniques,**Dimensionality Reduction Algorithms** : Principal Component Analysis (PCA),Linear Discriminant Analysis (LDA)

Unit-II: Supervised Learning Algorithms [6 Hours]

Linear Models : Linear Regression **:** Simple Linear Regression,Multiple Linear Regression , Polynomial Linear Regression, **Evaluation Metrics:** MAE, RMSE, R2,MSE,Logistic Regression,Ridge and Lasso Regression.

Kernel Based Algorithms: Support Vector Machine (SVM)- Linear Support Vector Machines, Linear Classification, Kernel based classification,**Probability-Based Algorithms :** Naive Bayes' Classifiers, Multinomial Naïve Bayes, and Gaussian Naive Bayes.

Unit-III : Supervised Learning : Classification [4 Hours]

K-nearest neighbor classifier,Decision Tree, **Ensemble Learning**: Bagging, Boosting, Random Forest, Adaboost, **Evaluation Metrics and Score**: Accuracy, Precision, Recall, Fscore, Cross-validation, Micro-Average Precision and Recall, Micro-Average F-score, Macro-Average Precision and Recall,Macro-Average F-score.

<u>Section 2: Topics/Contents</u>

Unit-IV: Introduction to Deep Learning [5 Hours]

Introduction, Evolution of AI, Machine Learning vs Deep Learning, Deep Learning types, Stages in ML/DL project, Applications of Deep Learning, Introduction to DL Frameworks Keras, PyTorch, Caffe, Shogun.**Basic Tensor Operations**: Creating, manipulating, and visualizing tensors. **Building a Neural Network**: Step-by-step implementation using Keras and PyTorch.

Unit-V: Convolution Neural Network [5 Hours]

CNN architecture overview, Building blocks of Convolutional Network-Convolution, activation functions (ReLU), pooling, fully connected layers.Padding, Strides, Typical Settings, the Fully Connected Layers **Advanced Architectures:** LeNet-5, AlexNet, VGG-16, ResNet.

Training and Optimization: Training strategies, regularization, transfer learning.

Implementation of neural network for a case study,case study: Real time applications

Unit-VI: Recurrent Neural Networks(RNN) [5 Hours]

Sequence modeling: Recurrent nets RNN architecture, bidirectional RNNs, Challenges in training RNNs,Long Short Term Memory (LSTM) ,Vanishing and exploding gradient problem, Auto encoders, **Applications of RNNs**: Language modeling, speech recognition, machine translation.

List of Practical :

1. Perform PCA in dimension reduction of numerical data

   a. Pre-process the data through standardization.
   b. Perform PCA to reduce dimension.
   c. Construct the scree plot.
   d. Data visualization in lower dimensional representation.

2. Implement Simple and Multiple Linear Regression to predict continuous variables.
   a. Perform data preprocessing (handle missing values, feature scaling).
   b. Fit a **Simple Linear Regression** model on a dataset (e.g., predicting house prices).
   c. Extend to **Multiple Linear Regression** with multiple features.
   d. Evaluate models using **MSE**, **RMSE**, and **R² Score**.
   e. Visualize the regression line and predictions.
3. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Python ML library classes can be used for this problem.
4. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
5. Learn Decision trees for regression and classification problem
   a. Split the data set into training and test sets.
   b. Build the decision tree
   c. Check model performances on training and test data sets.
   d. Apply cost complexity pruning to overcome overfitting problem
   e. Apply Random Forest algorithm to overcome overfitting problem.
   f. Apply Ada-boost ensemble method on Decision stumps.
6. Build a Multiclass classifier using the CNN model. Use MNIST or any other suitable dataset.
   a. Perform Data Pre-processing
   b. Define Model and perform training
   c. Evaluate Results using confusion matrix

7. Convolutional neural network (CNN) Use any dataset of plant disease and design a plant disease detection system using CNN.
8. Use MNIST Fashion Dataset and create a classifier to classify fashion clothing into categories. Using CNN
9. Implementation of RNN model for Stock Price Prediction
10. Using LSTM for prediction of future weather of cities in Python
11. Implement a basic **RNN** for handling sequential data.
    a. Build an **RNN** for a time-series prediction task.
    b. Train on sequential data (e.g., stock prices).
    c. Evaluate the model using **MSE** or **RMSE** for regression tasks.
    d. Visualize predictions vs actual values over time.

12. Design an object detection model using deep neural networks for simple objects.
    a. Select appropriate dataset and perform data pre-processing
    b. Define architecture in terms of layers
    c. Evaluate Model performance Label the object with appropriate text.
13. Train a CNN model using data augmentation to improve generalization.
    a. Apply data augmentation techniques like rotation, zooming, and flipping.
    b. Train the CNN on augmented data and compare performance with the original model.
    c. Plot training and validation accuracy for both models.

List of Seminar Topics:

1. Explainable AI (XAI): Making Machine Learning Models Transparent
2. Reinforcement Learning in Autonomous Systems
3. Federated Learning: Collaborative Learning Without Centralized Data
4. Classification of skin cancer with deep neural networks
5. Self-Supervised Learning: Closing the Gap Between Supervised and Unsupervised Learning

6.Convolutional Neural Networks (CNNs) in Image Processing

7. Accelerating Deep Network Training by Reducing Internal Covariate Shift
8. Deep learning applications for predicting pharmacological properties of drugs
9. GAN (Generalized Adversarial network)
10. Auto encoders
11. LSTM

List of Course Group Discussion Topics:

1. Supervised vs. Unsupervised Learning: When to Use Which Approach?
2. The Role of Data Quality in Machine Learning: Can Good Data Outperform Advanced Algorithms?
3. Reinforcement Learning vs. Traditional Machine Learning
4. Hyperparameter tuning: Is there a rule of thumb?
5. Deep Learning vs. Traditional Machine Learning

6 Which cost function: Least squared error or binary cross entropy?

7. Convolutional Neural Networks (CNNs) vs. Recurrent Neural Networks (RNNs): Strengths and Weaknesses
8. Need of hundred classifiers to solve real world classification problem
9. Which optimization: Batch gradient descent of stochastic gradient descent
10. Data Privacy in the Age of AI: Machine Learning and Deep Learning in Sensitive Domains
11. The Role of Data Augmentation in Enhancing Deep Learning Models

List of Design based Home Assignments:

Design:

1. Design a Machine Learning Model for Predicting Housing Prices

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
6. Design an LSTM for Machine Translation

Case Study:

1. Credit Risk Scoring for Banking
2.Churn Prediction for Telecom Industry
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

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Lab Examoination: End Semester Examination (Written): 100 Marks converted to 50 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

Text Books:

1. T. Mitchell, “Machine Learning”, McGraw-Hill, 1997.
2. Anup Kumar Srivastava, Soft Computing, Alpha Science International limited. 2009.
3. Goodfellow, I., Bengio, Y., and Courville, A., Deeep Learning, MIT Press, 2016.

4. C., M., Pattern Recognition and Machine Learning, Springer, 2006.

Reference Books:

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.
4. Richard S. Sutton and Andrew G. Barto, “Reinforcement Learning: An Introduction” by SethWeidman, “Deep Learning from Scratch: Building with Python from First Principles”O’Reily
5. Francois Duval, “Deep Learning for Beginners, Practical Guide with Python and Tensorflow”

MOOCs Links and additional reading material:

1. www.nptelvideos.in
2. https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs11
3. https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs50
4.https://www.my-mooc.com/en/categorie/deep-learning

Course Outcomes:

The student will be able to –

1) Understand the role of machine learning, data preprocessing techniques, including feature selection, PCA, and LDA, to improve model performance.
2) Apply linear models such as Linear Regression and Logistic Regression, and kernel-based models such as SVM for classification tasks.
3) Apply ensemble techniques like Bagging and Boosting, and classification algorithms such as KNN and Decision Trees to solve classification problems.
4) Apply a basic convolutional neural network using a deep learning framework
5) Evaluate the performance of CNN models and analyze the benefits of transfer learning using pretrained models in specific applications.
6) Analyze the strengths and weaknesses of RNNs in handling long-term dependencies in sequence modeling tasks.

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.

Laboratory Continuous Assessment: 100 Marks converted to 10 Marks
Course Project: End Semester Examination: 100 Marks converted to 20 Marks
Lab Examoination: End Semester Examination (Written): 100 Marks converted to 50 Marks
Comprehensive Viva Voce: End Semester Examination: 100 Marks converted to 20 Marks

Text Books:

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:

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:

1. www.nptelvideos.in
2. https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs11
3. https://nptel.ac.in/noc/courses/noc20/SEM1/noc20-cs50

Course Outcomes:

The student 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:

	Program Outcomes (PO)												PSO			
CO/PO	PO 1	PO2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	
CO1	3	3	3	2	3	1		1	1	1		2		3	3	
CO2	3	3	3	2	3	2		1	1	1		2		3	3	
CO3	2	3	3	3	3	2		1	1	1		2		3	3	
CO4	3	3	3	3	3	2		1	1	1		2		3	3	
CO5	3	3	3	3	3	2		1	1	1		2		3	3	
CO6	3	3	3	3	3	2		1	1	1		2		3	3	
Average	**3**	**3**	**3**	**2.66**	**3**	**1.83**		**1**	**1**	**1**		**2**		**3.0**	**3.0**	

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 Course 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, Data Analytics