Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Structure & Syllabus

B.Tech. (Computer Engineering)

With Effect from Academic Year 2024-25

Prepared by: - Board of Studies in Computer Engineering

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

Chairman – BOS **Chairman – Academic Board**

Vision of the Institution

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

Mission of the Institution

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

Vision of the Department

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

Mission of the Department

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

List of Programme Outcomes [PO]

PO	PO 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.

List of PSO Statements

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

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

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

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

Program Educational Objectives (PEOs)

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

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	CS
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 **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	CS2235	PCC: Computer Organization and Architecture	2	0	0	-	35	-	35	-	-	-	-	30		100	2
S6	CS2229	CC: Design Thinking-3	-	-	-	-	-	-	-	-	-	-	-	-	100	100	1
S7	CS2242	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**	**295**	**700**	**24**

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

Branch: Computer **Year:** S.Y. **A.Y.:** 2024-25 **Module:** IV

Sub 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 (%)	Test (%)	HA (%)	LAB (%)	CP (%)	PPT /GD (%)	CVV (%)	ESE (%)		
S1	CS2065	PCC: Data Structures	2	2	1	10	-	-	-	-	50	20	-	20		100	4
S2	CS2008	PCC: Operating Systems	2	2	1	10	-	-	-	-	50	20	-	20		100	4
S3	CS2245	MD: Microprocessors & Microcontrollers	2	2	1	10	-	-	-	20	-	20	-	20	30 (WRT)	100	4
S4	CS2246	PEC: Computer Graphics and Virtual Reality	2	2	1	10	-	-	-	-		20	20 (PPT)	20	30 (WRT)	100	4
S5	CS2247	ESC: Theory of Computation	2	-	-	-	35	-	35	-	-	-	-	30		100	2
S6	CS2229	CC: Design Thinking-3	-	-	-	-	-	-	-	-	-	-	-	-	100	100	1
S7	CS2242	PCC: Engineering Design and Innovation-	-	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 **Year:** T.Y. **A.Y.:** 2024-25 **Module: V**

Subject No.	Subject 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	CS3052	Computer Networks (Modules V & VI))	2	2	1	10	-	20	-	20	-	20	30 (WRT)	100	4
S2	CS3205	Design and Analysis of Algorithms	2	2	1	10	-	-	-	20	20 (PPT)	20	30 (WRT)	100	4
S3	CS3215	Web Technology	2	2	1	10	-	-	50	20	-	20	-	100	4
S4	CS3061	Software Modelling and Design	2	2	1	10	-	-	-	20	20 (GD)	20	30 (MCQ)	100	4
S5	CS3059	Design Thinking-5	-	-	-	-	-	-	-	-	-	-	100	100	1
S6	CS 3060	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**	**20**	**30**	**40**	**80**	**60**	**40**	**80**	**200**	**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

3. Ab Initio-Data Engineering ----offered by Ab Initio

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

Branch: Computer **Year:** T.Y. **A.Y.:** 2024-25 **Module:VI**

Subject No.	Subject 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	Coursera*	Coursera*	-	-	-	-	-	-	-	-	-	-	100	100	4
S2	CS3202	Artificial Intelligence	2	2	1	10	-	-	50	20	-	20	-	100	4
S3	CS3220	Cyber Security	2	2	1	10	-	-	-	20	20 (PPT)	20	30 (WRT)	100	4
S4	CS3053	Compiler Design	2	2	1	10	-	20	-	20	-	20	30 (MCQ)	100	4
S5	CS3059	Design Thinking-5	-	-	-	-	-	-	-	-	-	-	100	100	1
S6	CS 3060	Engineering Design and Innovation - 3	-	12	-	-	30	-	-	-	-	-	70	100	6
S8	SH3001	Reasoning and Aptitude Development	-	-	-	-	-	-	-	-	-	-	-	-	1
S9	AC*	Audit Courses*	-	-	-	-	-	-	-	-	-	-	-	-	0
		Total	**6**	**18**	**3**	**30**	**30**	**20**	**50**	**60**	**20**	**60**	**330**	**600**	**24**

Coursera*

Subject Code	Subject Name	Subject Code	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		

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

Branch: Computer **Year:** BTech **A.Y.:** 2024-25 **Module: VII(Course Work)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
OE1	Coursera*	Coursera Courses*								100			100	4
OE2	CS4275 Swayam	Introduction to Machine Learning	2	-	-	10	-	30	-	30	-	30	100	2
OE2	CS4276 Swayam	Deep Learning	2	-	-	10	-	30	-	30	-	30	100	2
Major Project	CS4225	Major Project	0	20	-	-	-	30	-	70	-	-	100	9
-	------------	Design Thinking-5	-	-	-	-	-	-	-	100	-	-	100	1
		Total	**4**	**20**	-	**20**	-	**90**	-	**330**	-	**60**	**500**	**16**

Coursera Courses*

Subject Code	Subject Name	Subject Code	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		

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

Branch: Computer **Year:** BTech **A.Y.:** 2024-25 **Module: VIII (Course Work)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
OE1	MD4202	Project Management	2	0	-	10	-	30	-	30	-	30	100	2
	LL4001	Generative AI	2	0	-	10	-	30	-	30	-	30	100	2
OE2	CS4217	Human Computer Interaction	2	-	-	10	-	30	-	30	-	30	100	2
	CS4272	Neural Networks	2	-	-	10	-	30	-	30	-	30	100	2
	CS4222	Image Processing	2	-	-	10	-	30	-	30	-	30	100	2
	ET4230	Natural Language Processing	2	-	-	10	-	30	-	30	-	30	100	2
OE3	CS4275 Swayam	Introduction to Machine Learning	2	-	-	10	-	30	-	30	-	30	100	2
	CS4276 Swayam	Deep Learning	2	-	-	10	-	30	-	30	-	30	100	2
Major Project	CS4225	Major Project	0	20	-	-	-	30	-	70	-	-	100	10
		Total	**6**	**20**	-	**30**	-	**120**	-	**160**	-	**90**	**400**	**16**

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

Branch: Computer **Year:** BTech **A.Y.:** 2024-25 **Module: VII (Internship)**

Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
OE1	Coursera*	Coursera Courses*								100			100	4
S1	CS4232	Industry Internship	-	32	-	-	-	30	-	70	-	-	100	11
S1	CS4234	International Internship	-	32	-	-	-	30	-	70	-	-	100	11
S1	CS4202	Research Internship	-	32	-	-	-	30	-	70	-	-	100	11
S2	------------	Design Thinking-5	-	-	-	-	-	-	-	100	-	-	100	1
		Total		**32**	-	-	-	**30**	-	**270**	-		**300**	**16**

Coursera Courses*

Subject Code	Subject Name	Subject Code	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		

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

Branch: Computer **Year:** BTech **A.Y.: 2024-25** **Module: VIII (Internship)**

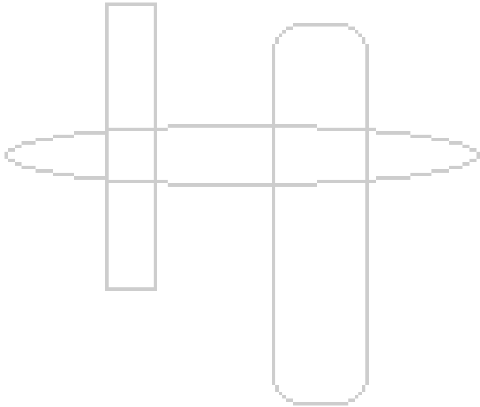
Subject No.	Subject Code	Subject Name	Teaching Scheme (Hrs/Week)			Examination Scheme								Credits
			Th	Lab	Tut.	CA		MSA		ESA			Total	
						HA (%)	LAB (%)	MSE (%)	PPT (%)	ESE (%)	GD (%)	Viva (%)		
S1	CS4232	Industry Internship	-	32	-	-	-	30	-	70	-	-	100	16
S1	CS4234	International Internship	-	32	-	-	-	30	-	70	-	-	100	16
S1	CS4202	Research Internship	-	32	-	-	-	30	-	70	-	-	100	16
		Total	**-**	**32**	**-**	**-**	**-**	**30**	**-**	**70**	**-**	**-**	**100**	**16**

INDEX

SN	Particular	Page No
1	Second Year Module -III Content	02
2	Second Year Module -IV Content	34
3	Third Year Module -V Content	61
4	Third Year Module -VI Content	89
5	Final Year Module -VII & VIII Content	117
6	Assessment Guidelines AY 2024-25	150

S. Y. B. Tech. Computer Engineering
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	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
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.Describetheproblem 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. Makeanexhaustivelistofeverythinginside,justoutside(immediatesurroundings)andon 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	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
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	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
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	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
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**

CS2235 :: Computer Organization and Architecture

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

Course Prerequisites: Basics of computer systems

Course Objectives:

1.Understand computer evolution, components, interconnection structures, bus interconnection and apply various binary arithmetic algorithms on signed as well as unsigned numbers.
2.Summarize working of the Central Processing Unit and RISC and CISC Architecture.
3.Outline the hardwired control unit design methods and control unit.
4.Design memory with due consideration of trade-offs and performance issues.
5.To analyze a pipeline for consistent execution of instructions with minimum hazards
6.To interpret the Input / Output transfer techniques and discuss the working mechanisms of various I/O peripherals

Course Relevance:

The field of Computer Organization is fundamental to understanding the inner workings of computers and how their various components interact and perform tasks. This course is designed to provide a comprehensive overview of computer systems, covering essential topics from the evolution of computers to advanced concepts like pipelining and input/output organization.

Section 1: Topics/Contents

Unit 1: Computer Evolution & Arithmetic **(5 hours)**

Organization & Architecture, Structure & Function, Brief History of Computers, Integer Representation: Fixed point & Signed numbers. Integer Arithmetic: 2’s Complement arithmetic, multiplication, Booth’s Algorithm, Division Restoring Algorithm, Non-Restoring algorithm.

Unit 2: Processor Design **(4 hours)**

CPU Architecture, Register Organization, Instruction types, Types of operands, Types of operation, Instruction formats, Addressing modes and address translation. Instruction cycles. RISC Processors: RISC- Features, CISC Features, Comparison of RISC & CISC Processors. Case study of Processor: Von Neumann Architecture.

Unit 3: Control Unit **(4 hours)**

Fundamental Concepts: Single Bus CPU organization, register transfers, performing an arithmetic/ logic operation, fetching a word from memory, storing a word in memory, Execution of a complete instruction. Micro-operations, Types of Control Unit, Hardwired Control, Micro-programmed Control: Microinstructions.

Section 2: Topics/Contents

Unit 4: Memory Organization (5 hours)

Need, Hierarchical memory system, Characteristics, Size, Access time, Read Cycle time and address space. Main Memory Organization: ROM, RAM, EPROM, E^2PROM, DRAM, Cache memory Organization, Cache Mapping techniques: Direct, Set Associative, Fully Associative.

Unit 5: Pipelining (5 hours)

Data hazards: operand forwarding, handling data hazards in software, side effects. Instruction hazards: unconditional branches, conditional branches, and branch prediction. Performance considerations: effect of instruction hazards, number of pipeline stages.

Unit 6: I/O Organization (5 hours)

Input/output systems, I/O Transfer Techniques: Program-controlled, Interrupt-Driven, DMA-controlled synchronous, asynchronous, working mechanisms of peripherals: keyboard, video displays, touch screen panel, printers

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. William Stallings, "Computer Organization and Architecture: Designing for Performance", 7th Edition, Pearson Prentice Hall Publication, ISBN 81-7758-9 93-8.
2. C. Hamacher, V. Zvonko, S. Zaky, "Computer Organization", 5th Edition, Tata McGraw Hill Publication, ISBN 007-120411-3.
3. Kai Hwang, "Advanced Computer Architecture", Tata McGraw-Hill ISBN 0-07-113342-9

Reference Books: (As per IEEE format)

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

MOOCs Links and additional reading material:

1. www.nptelvideos.in
2. www.geeksforgeeks.org
3. www.udemy.com

Course Outcomes:

Students will be able to -

1. Understand the structure, function, and characteristics of computer systems.
2. Describe the working of the Central Processing Unit and RISC and CISC Architecture.
3. Explore the knowledge about Control Unit Design.
4. Design memory with due consideration of trade-offs and performance issues.
5. Analyze a pipeline for consistent execution of instructions with minimum hazards.
6. Interpret the Input / Output transfer techniques and discuss the working mechanisms of various I/O peripherals

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3						1			1			3		2	
2	2	2								1			2	3		
3	3	2		2		1	1						2	2	2	
4			3	2		2	3		2				3		3	
5		2	3	2	3							2		3	2	
6	3		2	1						2		2				2
Avg	**2.75**	**2.0**	**2.66**	**1.75**	**3.0**	**1.5**	**1.67**	**0.0**	2.0	**1.33**	**0.0**	**2.0**	**2.5**	**2.67**	2.25	**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: Operating Systems, Compiler Design, Systems Programming

Job Mapping:

Computer Organization is necessary part of any core computer engineering job. Without having sound knowledge of Computer Organization, it is not possible to be good in Computer Engineer.

Syllabus Template **FF No. : 654**

CS2229 and CS3059: Design Thinking 3 and 5

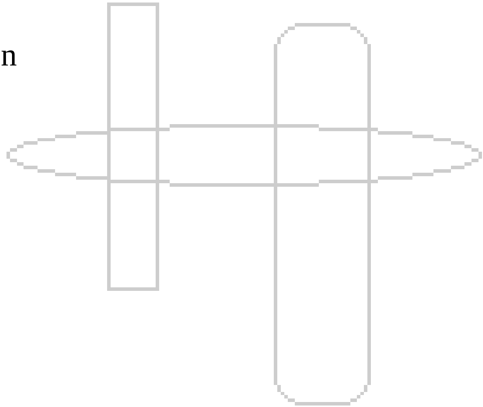
Credits: 01 **Teaching Scheme: Tutorial 01 Hr/week**

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

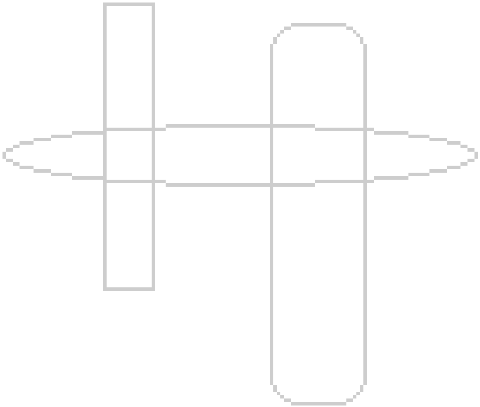
Structure of The paper
Journal List (Top 50 Journals)
Selection of the journal
Use of various online journal selection tools
Plagiarism checking
Improving contents of the paper
Patent drafting
Patent search
Filing of patent
Writing answers to reviewer questions
Modification in manuscript
Checking of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

	Program Outcomes (PO)												PSO			
CO/PO	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	1	1	1	1	1	--	--	--	--	--	--	1	1	2	2	3
CO2	1	1	1	1	1	--	--	--	--	--	--	1	2	1	1	3
CO3	2	2	3	3	2	2	1	2	2	3	--	1	2	2	3	3
CO4	3	3	3	3	3	2	1	2	2	3	1	1	-	-	2	3
CO5	1	1	1	1	1	--	--	--	--	--	--	1	-	-	1	2
CO6	2	2	2	2	2	2	1	3	2	3	--	1	2	2	2	3
C07	1	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**	1.66	1.66	1.71	2.5



Syllabus Template **FF No. : 654**

CS2242 and CS3060: 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, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
- This is the sample list to start with. Faculty and students are free to include other areas which meet the society requirements at large.
- The model begins with the identifying of a problem, often growing out of a question or “wondering”. This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students’ wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher’s Role in PCL :

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

Student's Role in PCL:

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

- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

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

Literature Survey – To avoid reinvention of wheel:

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

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

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

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

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

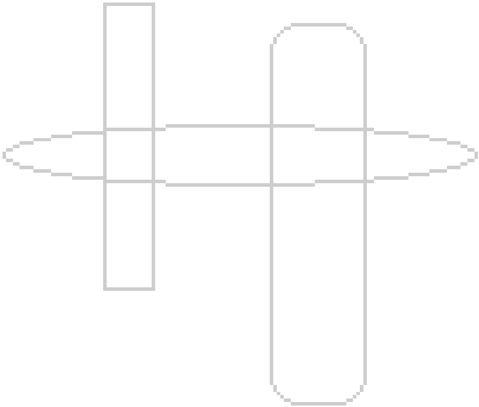
	Program Outcomes (PO)												PSO			
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	2	2	2					3		2	2				3
CO2	2	2	3	2	2		2		3		2	2	3		3	3
CO3	2	2	3	2	3		2		3		2	2	3		3	3
CO4	2	2	3	2	3	3		2	3		2	2	3	3	3	3
CO5	2	2	3	2	3	2			3		2	2	3		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**	**3.0**	**3.0**	**3.0**	**3.0**

CO attainment levels

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

S. Y. B. Tech. Computer Engineering
AY 2024-25

Module IV Course Content



Syllabus Template **FF No. : 654**

CS2065:: 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, Leafnode 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**

CS2008:: 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. 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	PSO4
1	2	2												3	3	
2	2	2	3	2			2	2						3	3	
3	2	3	3	2			2	2						3	3	
4	2	3		2										3	3	
5	2	3		1										3	3	
6	2	2	3	2		2	2	2	3	3.0	3.0	3.0		3	3	2.0
Avg	2	2.5	3.0	1.8		2.0	2.0	2.0	3.0	3.0	3.0	3.0		3.0	3.0	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

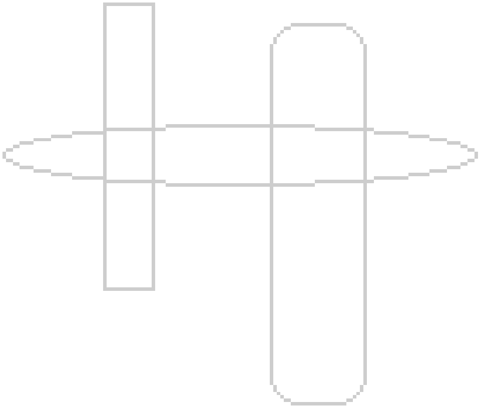
CO1 – L1, CO2 – L2, CO3 – L3, CO4 – L4, CO5 – L4 and CO6 – L5

Future Course Mapping:

Advanced Operating System, Distributed Operating System, Parallel Computing.

Job Mapping:

System Administrator, System Analyst



Syllabus Template **FF No.: 654**

CS2245: Microprocessor & Microcontroller

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

Course Prerequisites:

1. Microprocessors and Interfacing Ttechniques
2. Computer Organization and Architecture.

Course Objectives:

1. To study the fundamental concepts of Computer System and Microprocessor.
2. To gain knowledge of Processor Operating Modes.
3. To describe data organization of Pentium processor.
4. To analyse the functioning of Microprocessor.
5. To get familiar with Microprocessor and Microcontroller architecture.
6. To find the solutions for real world problems of processing.

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. Microprocessor and Microcontroller is a crucial part of Computer System Hardware as they allow designer to design system efficiently. A wise selection of Microprocessor and Controller can improve the overall performance of a System.

Section 1: Topics/Contents

Unit-I Introduction to Microprocessors **5 Hours**

Block Architecture, Microprocessor Operations (read, Write etc), Hardware & Software Requirement. Introduction to 8086, Block diagram of 8086, Segmentation in 8086,Addressing modes of 8086

Unit-II Operating Modes of Microprocessor **5 Hours**

Introductionto 80386,Block diagram of 80386,Segmentation and Paging in 80386

Introduction to Pentium Processor, Real Mode, Protected Mode, Software Model of Pentium, Functional Description of Pentium, Register Set of Pentium.

Unit-III Data Organization of Pentium **5 Hours**

Pentium Data Organization, Instruction Types, Addressing Modes, Interrupts.

Section2: Topics/Contents

Unit-IV Hardware Details of Pentium **5 Hours**

Introduction, CPU Pin Description, RISC Concepts, Bus Operations, The Pentium’s Superscalar Architecture, Pipelining, Branch Prediction, The Instruction & Data Caches, The Floating-Point Unit.

Unit-V Introduction to Microcontroller **4 Hours**

Comparison of Microprocessor & Microcontroller, Block and Functional Diagram of ARM Controller, Pin Configuration of ARM Microcontroller.

Unit-VI ARM Data Organization 4 Hours

Instruction Set of ARM Microcontroller, Addressing Modes, Data Types, Register Set, Applications of ARM Microcontroller.

List of Tutorials:

1. Microprocessor Operation (M/IO read and write)
2. Hardware and Software Requirement of Microprocessor Based System
3. Data organization of Pentium.
4. Addressing modes of Pentium
5. Instruction Set of Pentium.
6. Data organization of ARM.
7. Addressing modes of ARM
8. Instruction Set of ARM.
9. Microprocessor Based System
10. Microcontroller Based System.

List of Practical's (Minimum Six):

1. Write an ALP to find positive number and negative numbers from the array of signed number stored in memory and display magnitude of negative and positive numbers.
2. Write an ALP to perform non-Overlapping block transfer operation without using string operations, Data bytes in a block stored in one array transfer to another array. Use debugger to show execution of program.
3. Write an ALP to convert 4-digit HEX number into equivalent 5-digit BCD number
4. Write an ALP to convert 5-digit BCD number into equivalent 4-digit HEX number.
5. Write an ALP for Multiplication of two 8-bit Binary numbers using Successive and Add and Shift Method.
6. Write an ALP for following operations on the string entered by the user (Use Extern Far Procedure).
    a. Concatenation of two strings
    b. Find number of occurrences of sub string in the given string
    c. Find number of occurrences words and lines.
7. Write an ALP for following operations on the string entered by the user.
    a. String length
    b. Reverse of the String
    c. Palindrome
8. Write an ALP to encrypt and decrypt the given message.
9. Write an ALP to perform following operations on file.
    a. Open File
    b. Write data in the file.

c. Delete data in the file.
d. Close the file.

List of Course Project Areas:

1. Weather Imaging CubeSat with Telemetry Transmission.
2. Ebike Speed Controller System.
3. Air Water Pollution Sensing Smart Watch.
4. Solar Sea Weather and Pollution Transmitter Buoy.
5. Coin Operated Water ATM with Bottle Dispenser.
6. Multiple Cities Load Shedding Using ARM
7. Wireless Biomedical Parameter Monitoring System Using ARM9
8. ARM and RFID Based Security System (Home, Office, Industrial)
9. Advanced Electronic Voting Machine (EVM) using ARM)
8. Online Parallel Examination.
9. Machine Learning, Deep Learning, AI, Blockchain etc Based
10. Agriculture, Health Care, Education, Govt., Transportation, Banking, Insurance Based but not limited for.

List of Seminar Topics:

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

List of Design based Home Assignments:

1. Write the sequence of control steps required for the single bus organization for each of the following instructions:

i. ADD the (immediate) number NUM to register R1

ii. ADD the contents of memory location NUM to register R1

Assume that each instruction consists of two words. The first word specifies the operation and addressing mode, and second word contains the number NUM

2. Configure a 32 Mb DRAM chip. Consider cells to be organized in 8K X 4 array. Find out the number of address lines.

3. A set associative cache consists of 64 lines, or slots, divided into four-line sets. Main memory contains 4K blocks of 128 words each. Analyze the format of main memory addresses with proper explanation.
4. A one pipeline system takes 50 ns to process a task. The same task can be processed in 6 segmentpipelines with a clock cycle of 10 ns. Determine the speedup ratio of pipeline for 100 tasks. Whatis maximum speedup ratio?

List of Case Study based Home Assignments:

1. Micro-programmed Control Unit and Hardwired Control Unit.
2. Pipelining
3. Floating Point Unit of Pentium
4. Modern Microprocessors
5. Modern Microcontrollers

List of Blog based Home Assignments:

1. Super Computer
2. Intel Journey
3. New ARM Interconnect technologies
4. Distributed Systems and Parallel Computing
5. Intel I series microprocessor

List of Survey based Home Assignments:

1. New memory technologies and their potential impact on architecture
2. Virtual Memory
3. Simulation of a Superscalar processor and analyzing impact of design trade Offs
4. Cache Consistency Models in Modern Microprocessors
5. Flash Memory

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
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:

1. *James Antonakos , "The Pentium Microprocessor", 2004, Pearson Education ISBN – 81-7808-545-3*
2. *Muhammad Ali Mazidi and Janice Gillispie Mazidi, " The 8051 Microcontroller and embedded systems", 2009, Pearson education, ISBN – 81-7808-574-7*
3. *Intel 8-bit Microcontroller manual.*
4. Ajay Deshmukh, "Microcontrollers – (Theory and application)" , 2004, TMH,ISBN 0-07-058595-4

5. *Intel 8096 16-bit Microcontroller manual*.

Reference Books:

1.*The 8051 Microcontroller and Embedded systems using Assembly and C. (K.J.Ayala/ D.V.Gadre) -Cengage learning ISBN 9788131511053*

2. *Jeffry and Royer, "IBM PC Hardware and Assembly Language", BPB Publication*

3. *8051 Microcontroller Hardware, Software and applications. (V Udayshankara and M.S. Mallikarjunaswamy) (TMH)*

MOOCs Links and additional reading material:

1. https://onlinecourses.nptel.ac.in/noc22_ee12/preview
2. https://www.mooc-list.com/course/interfacing-arduino-coursera
3. https://www.my-mooc.com/en/mooc/embedded-systems-shape-the-world-microcontroller-input-output/
4. https://onlinecourses.swayam2.ac.in/cec21_cs16/preview
5. https://www.youtube.com/watch?v=RLJmQWQwBJw

Course Outcomes:

The student will be able to:

1. Describe the various microprocessor operations (1).
2. Illustrate operating modes of Pentium processor (2).
3. Correlate concept of data organization, instruction types & addressing modes of Pentium processor (2).
4. Demonstrate working model of Pentium processor (3).
5. Compare Microprocessor & Microcontroller (4).
6. Evaluate real world application of Microcontroller (5).

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3	2	2		-	-	-	-	-	-	-	2	-	3	-	-
2	3	2	2			-	-	-	-	-	-	2	-	3	-	-
3	3	2	3			-	-	-	-	-	-	2	3	3	-	-
4	3	3	2				-	-	-	-	-	2	-	3	-	-
5	3	2	3				-	-		-	2	2	-	3	-	-
6	3	2	2	2	2		2	2	2	-	2	2	3	3	3	-
Avg	3	2.16	2.33	2.0	2.0		2.0	2.0	2.0		2.0	2.0	3.0	3.0	3.0	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

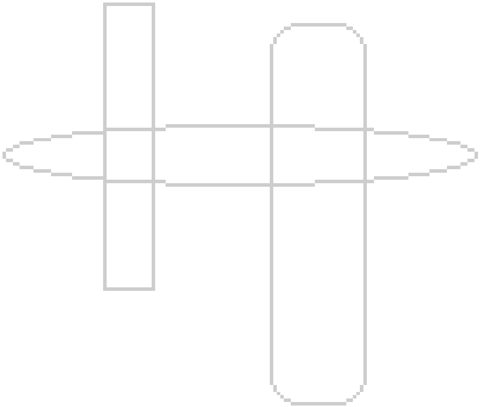
CO1 – L1, CO2 – L2, CO3 – L2, CO4 – L3, CO5 – L4 and CO6 – L5

Future Course Mapping:

Cloud Computing, Distributed System, Software Design and Modelling

Job Mapping:

Application Developers, System programmer



Syllabus Template **FF No. : 654**

CS2246: Computer Graphics & Virtual Reality

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

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites: Data Structures**,** Mathematics – Linear algebra is recommended. and good programming skills in C

Course Objectives:

1. To understand how the various elements, like algebra, geometry, algorithms and data structures interact in the design of graphics.
2. To analyze various algorithms to scan, convert the basic geometrical primitives, transformations, Area filling, clipping
3. To develop the programming and application skills in computer graphics.
4. To understand a typical graphics pipeline
5. To apply the graphics objects for the real usage in society and industries.
6. To Apply principles of computer graphics to specific problems in a variety of domains.

Course Relevance:

Computer graphics are an essential part of modern software. In this course students will learn about fundamental algorithms, data structures and programming models used in 3D graphics applications. These key concepts in computer graphics programming will be covered from their mathematical foundations through to their application in domains such as data visualization, virtual reality, computer games and film animation/VFX.

Section 1: Topics/Contents

Unit-I Title: Introduction to Computer graphics

Define fundamental parameters required for computer graphics, applications of computer graphics B-Spline curves, Bezier curves, interpolation & fractals, scan conversion of line.

Unit-II Title: Polygons and Clipping

Scan Converting Lines, Mid-point criteria, Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles, Scan Converting Ellipses, Filling Polygons, edge data structure, Clipping Lines algorithms– Cyrus-Beck, Cohen-Sutherland and Liang-Barsky, Clipping Polygons, problem with multiple components.

Unit-III Title: 2D Transformation

Transformations and Matrices, Transformation Conventions, 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Translations and Homogeneous Coordinates, Rotation, Reflection, Scaling, Combined Transformation, Transformation of Points, Transformation of The Unit Square, Solid Body Transformations, Rotation About an Arbitrary Point, Reflection through an Arbitrary Line, A Geometric Interpretation of Homogeneous Coordinates, The Window-to-Viewport Transformations. Inverse Transformations

Section2: Topics/Contents

Unit-IV Title: 3D Transformations and Projections

Introduction, Three-Dimensional Scaling, Three-Dimensional Shearing, Three-Dimensional Rotation, Three-Dimensional Reflection, Three-Dimensional Translation, Multiple Transformation, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Matrix Representation of 3D Transformations, Composition of 3D Transformations, Affine and Perspective Geometry, Perspective Transformations, Techniques for Generating Perspective Views, Vanishing Points, the Perspective Geometry and camera models, Orthographic Projections, Axonometric Projections, Oblique Projections, View volumes for projections.

Unit-V Title: Plane Curves and Surfaces

Curve Representation, Nonparametric Curves, Parametric Curves, Parametric Representation of a Circle, Parametric Representation of an Ellipse, Parametric Representation of a Parabola, Parametric Representation of a Hyperbola, A Procedure for using Conic Sections, The General Conic Equation; Representation of Space Curves, Cubic Splines, , Bezier Curves, B-spline Curves, B-spline Curve Fit, B-spline Curve Subdivision, Parametric Cubic Curves, Quadric Surfaces. Bezier Surfaces. Fractal line and Surface Generation

Unit-VI Title: Introduction to Virtual Reality

Virtual Reality : Basic Concepts , Classical Components of VR System , Types of VR Systems, . Input / Output Devices, Graphical Rendering Pipeline, Haptic Rendering Pipeline, Open GL rendering pipeline, Applications of Virtual Reality

List of Tutorials (13):

1. Draw line using scan conversion algorithms like DDA algorithm & Bresenham’s Line drawing algorithm
2. 2D transformations applied on object
3. Implement various area filling techniques
4. Compare boundary fill & flood fill algorithm & assess result generated by both techniques.
5. Apply line & polygon, text clipping algorithms to clip the object and make it ready for display purpose.

6. Implement viewing transformation in computer graphics
7. Implement 3D object representation, polygon surfaces, tables & Meshes
8. Apply3D transformations on object
9. Define basic concepts of virtual reality & various types of virtual reality like immersive, nonimmersive & fully immersive virtual reality 10. Basic Operations on Spatial Data Structures
11. Explain need of virtual reality in real world & different I/O devices required for Virtual Reality like HMD, various types of trackers etc 12. Operations on Concurrent Data Structures
13. Explain Modeling techniques in virtual world & Collision detection.

List of Practical's (Minimum Six):

1. Line Drawing and Circle Drawing.
2. Polygon filling algorithms.
3. Line clipping algorithm.
4. Polygon clipping algorithm.
5. 2D transformations
6. Applications of 2D transformations.
7. 3D transformations
8. Applications of 3D transformations
9. Hidden surface removal
10. Curves and fractals

List of Course Project areas

1. Rendering of Deep Water using Spectral Domain techniques in Real Time
2. World Generation using Model Synthesis
3. Rendering Biological Iridescence4. Modeling the real world problems using graphs and trees
5. Multi-User VR Driving Simulator Game
6. Molecular Dynamics Visualisation in VR
7. Separable Subsurface Scattering for Photo-realistic Real-Time Skin Rendering.
8. A Virtual Reality Driving Simulator
9. Procedural Modelling of Urban Areas
10. Real-time Adversarial Gaming with a Robot in Virtual Reality
11. Real-Time Point-Based Rendering

Seminars:

List of Course Seminar Topics

1. Principle Of Teaching Draft Geometry And Computer Graphics

2. Physical validation of simulators in Computer Graphics: A new framework dedicated to slender elastic structures and frictional contact
3. Development of Innovative Neurosurgical Operation Support Method Using Mixed-Reality Computer Graphics Adversarial Search Techniques
4. Computer Graphics teaching challenges: Guidelines for balancing depth, complexity and mentoring in a confinement context
5. Single-shot fringe projection profilometry based on Deep Learning and Computer Graphics
6. A Computer Graphics-Based Framework for 3D Pose Estimation of Pedestrians
7. Types Of Computer Graphics And Their Practical Importance In Human Life
8. Cross-sectional design: A tool for computer graphics and computer-aided design courses
9. Novel Parallelization Techniques for Computer Graphics Applications
10. Deep Graphics Encoder for Real-Time Video Makeup Synthesis from Example
11. New perspectives for computer graphics
12. A Computer Graphics Simulation of Fire

List of Home Assignments:

List of Design Based Home Assignments

1. Use Of 3d Computer Modeling In Learning Engineering Graphics with example
2. Real-Time Simulation of a Cerebellar Scaffold Model on Graphics Processing Units
3. Efficient Rendering of Ocular Wavefront Aberrations using Tiled Point-Spread Function Splatting
4. Design and Visualization for Exploring Real-World Data
5. Identification and Classification of Off-Vertex Critical Points for Contour Tree Construction on Unstructured Meshes of Hexahedra
6. Implementation and evaluation of 3D graphics compression for optimizing the quality of user experience in networked virtual reality
7. Visualization of Tensor Fields in Mechanics
8. Segmentation and Recognition of Offline Sketch Scenes Using Dynamic Programming
9. Education Quality Issues in Engineering Graphics Training
10. Design of a Pupil-Matched Occlusion-Capable Optical See-Through Wearable Display

List of Case Study Based Home Assignments

1. Case study on Introducing the IEEE Virtual Reality
2. Case study on EllSeg: An Ellipse Segmentation Framework for Robust Gaze Tracking
3. Case study on a Class of “Basic Media Design” Based on 3D-CAD for Beginners of Digital Processing
4. Case study on GestOnHMD: Enabling Gesture-based Interaction on Low-cost VR Head-Mounted Display

5. A case study on acceleration Structure Designed for Moving PrimitivesChallenges/Issues in AI applications
6. A case study on Efficient Muscle Simulation in Deformation Space using graphics and VR
7. A case study on Wearable 3D Machine Knitting: Automatic Generation of Shaped Knit Sheets to Cover Real-World ObjectsFuture of AI

List of Blog Based Home Assignment

1. Modeling Soft Swimming Robots using Discrete Elastic Rod Method
2. Real-Time Geometric Glint Anti-Aliasing with Normal Map Filtering
3. A Framework for Data Visualization based on Particle Systems
4. Progressive path tracing with bilateral-filtering-based denoising
5. Virtual reality, the new 3-D interface for geographical information systems
6. Robust pencil drawing generation via fast Retinex decomposition
7. Automated 3D solid reconstruction from 2D CAD using OpenCV
8. Image Inpainting Using Double Discriminator Generative Adversarial Networks
9. 3D Manipulations with 2D Devices
10. Measuring presence in augmented reality environments: design and a first test of a questionnaire

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
Presentation: 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. "Computer Graphics", D. Hearn, M. Baker, 2nd Edition, Pearson Education, 2002, ISBN 81-7808-794-4.

2. "Procedural Elements for Computer Graphics", D. Rogers, 2nd Edition, Tata McGraw-Hill Publication, 2001, ISBN 0-07-047371-4.

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

1. "Computer Graphics", S. Harrington, 2nd Edition, McGraw-Hill Publications, ISBN 0 - 07 -100472 -6.
2. "Computer Graphics Principles and Practice", J. Foley, V. Dam, S. Feiner, J. Hughes, 2nd Edition, Pearson Education, 2003, ISBN 81-7808-038-9

MOOCs Links and additional reading material:

1. www.nptelvideos.in,
2. www.geeksforgeeks.org

Course Outcomes: The student will be able to –

1. Understand computer graphics fundamentals
2. Utilize algorithms to draw, fill and perform clipping on various 2D objects.
3. Use mathematics to achieve 2D transformations on different 2D geometrical shapes
4. Systematically identify and solve numerical problems of 3D transformations and projections
5. Interpret the curves and fractals to represent graphics system.
6. Understand basic concepts of virtual reality & various types of virtual reality like immersive, nonimmersive & fully immersive virtual reality to model real life applications

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	2	2			-	-	-	-	-	-	-	2	3		-	-
2	2	3	3			-	-	-	-	-	-	2	3		-	-
3	3	3	3			-	-	-	-	-	-	2	3		-	-
4	3	3	3	1			1	-	-	-	-	2	3	3	-	-
5	3	3	3	1			1	-		-		2	3	3	-	-
6	3	3	3	1	3	2	2	2	2	2.0	2	2	3	3	2	2
Avg	2.66	2.83	3.0	1.0	3.0	2.0	1.33	2.0	2.0	2.0	2.0	2.0	3.0	3.0	2.0	2.0

CO Attainment levels:

CO1 -2,   CO2 –3, CO3-3,   CO4-2,   CO5 -3,   CO6-3

Future Course Mapping:

Following courses can be learned after successful completion of this course: Advanced Computer Graphics, Image Processing, AR/VR, and Animation etc.

Job Mapping:

Computer Graphics and Virtual Reality is a necessary part of any core design, animation, & Vision, programming job. All Industries always looks for a strong knowledge in Computer Graphics and Virtual reality.

Syllabus Template **FF No. : 654**

CS2247::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**	**PSO4**
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										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**		**0**	**0**	**2**	**3**	**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

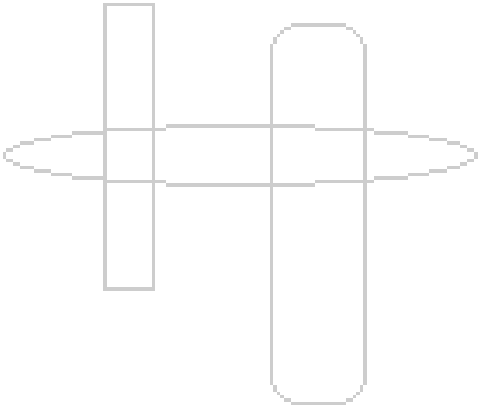
CO1 – L2, CO2 – L3, CO3 – L3, CO4 – L4, CO5 – L5 and CO6 – L5

Future Course Mapping:

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

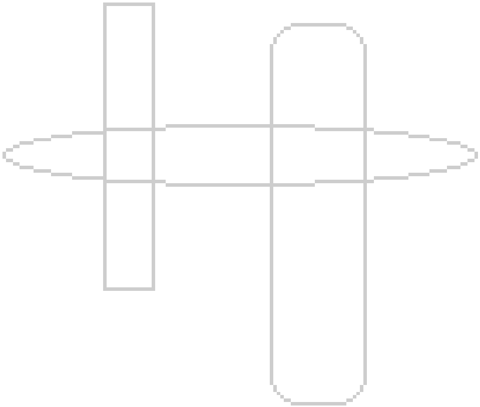
Job Mapping:

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 Engineering
AY 2024-25

Module V Course Content



Syllabus Template **FF No. : 654**

CS3052::Computer Networks

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

Course Prerequisites: Operating System, Theory of Computer Science

Course Objectives:

1. To learn the data communication model, signal generation, data encoding, digital modulation and demodulation required for wired and wireless communication networks.
2. To learn the physical layer which includes transmission mediums, physical layer devices, transmission modes and topologies, performance issues for intranet and internetworks.
3. To learn multiple access schemes and wide area network connectivity for intranet and internetworks.
4. To learn IP protocol and routing algorithms for packet switching service framework used in intranet and internetworks.
5. To learn TCP and UDP protocol to provide quality of service over packet switching service framework used for intranet and internetworks.
6. To learn to select, design, develop, analyze and evaluate client server solutions for societal requirements at large.

Course Relevance:

The key technology of the information age is communications. Data communications and networking is a truly global area of study, both because the technology enables global communication over telephone lines and Internet. Data communication and networking is the backbone of all IT infrastructures in the world. These technologies and applications often emerge in communication within countries of countries and spread rapidly around the world.

Section 1: Topics/Contents

Unit-I Data Communication, Networking Fundamentals, Physical Layer 04 Hours

Communication Model: Source, Transmitter, Transmission System, Receiver, Destination, Data Terminal Equipment (DTE), Data Communication Equipment (DCE). Transmission Configurations: Point to Point and Multipoint. Transmission Modes: Synchronous and Asynchronous. Transmission Methods: Serial and Parallel. Communication Modes: Simplex, Half Duplex, and Full Duplex. Line Coding: Unipolar NRZ, Polar NRZ, NRZ Inverted, Bipolar Encoding, Manchester Encoding, Differential Manchester Encoding. Modulation: Analog Modulation: Amplitude, Frequency, Phase.

Pulse Modulation Techniques: PCM, PAM, PWM, PPM. Digital Modulation: ASK, FSK, MSK, GMSK, PSK, BPSK, PSK, QAM, CPM, OFDM and multicarrier modulations.
Networking Fundamentals: Types of Computer Networks: LAN, MAN, WAN, PAN, Internet, internet and Intranet. Network Architectures: Client-Server; Peer To Peer. Network Architecture Modes: Infrastructure and Ad-hoc mode. Network Topologies: Mesh, Star and Hierarchical. Reference Models: OSI, TCP/IP. Design Issues for Layers. Is ATM still used? Is ISDN dying? Is Frame Relay outdated? Is SNA still present in the Market?
Physical Layer: Transmission Mediums: Air, Water, Vacuum, Coaxial, 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 Logical Link Control 06 Hours

Logical Link Control: Design Issues: Services to Network Layer, Framing, Error Control: Parity Bits, Hamming Codes and CRC. Flow Control Protocols: Unrestricted Simplex, Stop and Wait, Sliding Window Protocol, WAN Connectivity: PPP and HDLC. PPPoE, PPPoA. Is DOCSIS used in 2023? Do we use DSL line in 2023? Do we use coaxial cable in 2023? Is PPP still used?

Unit-III Medium Access Control 04 Hours

Medium Access Control: Channel Allocation: Static and Dynamic, Multiple Access Protocols: Pure and Slotted ALOHA, CSMA, WDMA. Legacy Standard : 10 Mbps IEEE 802.3 Standard(Ethernet), Wiring Schemes and Frame Formats, CSMA/CD, Binary Exponential Back-off Algorithm. High Speed Ethernet Standards: Fast, Gigabit and 10Gigabit (Focus must be on Gigabit Networks). Wireless Standards: Radio Spectrum, Frequency Hopping (FHSS) and Direct Sequence (DSSS), IEEE 802.11a/b/g/n/ac, IEEE 802.15, IEEE 802.15.4 and IEEE 802.16 Standards, CSMA/CA, Introduction of Infrastructure and Data Processing Unit (IPU and DPU)

Section2: Topics/Contents

Unit-IV Network Layer 06 Hours

Network Layer: Switching Techniques: Circuit, Message and Packet Switching. Logical Addressing: IPv4 and IPv6, Subnetting, NAT, CIDR. Network Layer Protocols: IP, ICMP, Routing Protocols: Distance Vector, Link State, and Path Vector. Routing in Internet: RIP, OSPF, BGP, Congestion control and QoS, MPLS, Mobile IP, Routing in MANET: AODV, DSR

Unit-V Transport Layer 04 Hours

Transport Layer: Services: Berkeley Sockets, Addressing, Connection Establishment, Connection Release, Flow control and Buffering, Multiplexing. HTH Layer Protocols: TCP, TCP, TCP Timer management, UDP. Quality of Service: TCP Congestion Control. Traffic Shaping: AIMD. QUIC Protocol, Real Time Support Protocols: Real Time Transport protocol (RTP), Stream Control

Transmission Protocol (SCTP), Quality of Service (QoS), Differentiated services, TCP and UDP for Wireless

Unit-VI Application Layer **04 Hours**

Application Layer: Address Resolution: Domain Name System (DNS). WWW: Hyper Text Transfer Protocol (HTTP1.1/1.2/2.0) and HTTPS with SSL. Web Service. Email: SMTP, MIME, POP3 and Webmail. File Transfer: FTP, Dynamic Logical Addressing: Dynamic Host Control Protocol (DHCP), Custom packet generation, Design, development and evaluation of scalable enterprise application using communication and service frameworks.

List of Tutorials (13):

Unit-I Data Communication, N/w Fundamentals and Phy Layer

1) Examples and analysis of Encoding Methods: Unipolar NRZ, Polar NRZ, NRZ Inverted, Bipolar Encoding, Manchester Encoding, Differential Manchester Encoding

2) Line coding, Channel Encoding and modulations Techniques: used in IEEE 802.3 standard and its extensions, IEEE 802.11 standards and its extensions for 100 Mbps, 1 GbE, 1 Gbps, 2.5 Gbps, 5 Gbps, 10 Gbps, 25Gbps, 40 Gbps, 100 Gbps networks. Chanel Encodings in 3G, 4G and 5G Mobile Networks

Unit-II Logical Link Control

3) Examples on Network Performance parameters: RTT, Delay, Bandwidth, Throughput and efficiency

Unit-III Medium Access Control:

4) PHY and MAC Layer IEEE 802.3 Standards For Copper: Overview of 10 Mbps Ethernet, Fast Ethernet, GbE -Gigabit Ethernet, 2.5 Gigabit Ethernet, 5 Gigabit Ethernet, 10 Gigabit Ethernet, 25 Gigabit Ethernet,40 Gigabit Ethernet, 100 Gigabit Ethernet

5) PHY and MAC Layer IEEE 802.3 Standards For Optical Fiber: 100 Mbps Fast Ethernet, GbE -Gigabit Ethernet, 2.5 Gigabit Ethernet, 5 Gigabit Ethernet, 10 Gigabit Ethernet, 25 Gigabit Ethernet,40 Gigabit Ethernet, 100 Gigabit Ethernet

6) PHY and MAC Layer IEEE 802.11 Wireless LAN Standards: IEEE 802.11, Wi-Fi 1/IEEE 802.11a, Wi-Fi 2/IEEE 802.11b, Wi-Fi 3/IEEE 802.11g, Wi-Fi 4/IEEE 802.11n, Wi-Fi 5/IEEE 802.11ac, IEEE 802.11ad (WiGig), IEEE 802.11ah (HaLow), Wi-Fi 6/IEEE 802.11ax, Wi-Fi 6/IEEE 802.11ay, Wi-Fi 6/IEEE 802.11by, Wi-Fi 7/IEEE 802.11be

Unit-IV Network Layer:

7) Examples of Network Layer Logical Addressing

(a) Classful IP and CIDR: Subnetting, IP Prefixes

(b) NAT Mapping: Public to Private IP and Port Mapping

(c) Packet Delivery in Internetwork: Packets traversing through different sub-networks with different MTU and Speeds

(d) Packet Dropping Probabilities of Routers

8) Examples of Network Layer Routing
(a) Shortest Path and Spanning Tree
(b) Dijkstra’s Algorithm
(c) Distance Vector Routing
(d) Link State Routing
(e) ECMP

Unit-V Transport Layer
9) Examples of Transport Layer
(a) TCP Connection Establishment: SYN and ACK, Normal Packets
(b) Flow Control: Calculating Optimal Size of Sliding Window
(c) Cumulative ACK scheme
(d) Smoothed RTT
(e) Slow Start and Additive Increase

Unit-VI Application Layer:
10) Examples of Application Layer
(a) DNS: URL Domain Processing
(b) Performance of HTTP1.0 and HTTP1.1
(c) CDN--------

List of Practical’s (Minimum Six):

Unit-I Data Communication Networking Fundamentals and Physical Layer:
1) Write a program in C++/JAVA to implement - Unipolar NRZ, Polar NRZ, NRZ Inverted, Bipolar Encoding, Manchester Encoding and Differential Manchester Encoding.
2) Setting up small computer networks and Hands on networking commands:
Set up a small wired and wireless network of 2 to 4 computers using Hub/Switch/Access point. It includes installation of LAN Cards, Preparation of Cables/ Installation and Configuration of Access Point, Assigning unique IP addresses and use of ping utility. Hands on for network commands - ping, pathping, ipconfig/ifconfig, arp, netstat, nbtstat, nslookup, route, traceroute/tracert, nmap.

Unit-II and III MAC and Logical Link Layer
3) Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.(50% students will perform Hamming Code and others will perform CRC). Further extend it to real implementation of CRC over Ethernet standard.
4) Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in peer to peer mode and demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode. Further extend it to real implementation of Flow Control over TCP protocol.

Unit-IV Network Layer

5) 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-V Transport Layer

6) Write a program using TCP Berkeley socket primitives for wired /wireless network for following
a. Say Hello to Each other
b. File transfer
c. Calculator (Arithmetic)
d. Calculator (Trigonometry)
Demonstrate the packets captured traces using Wireshark/Fiddler for traffic analysis tool in peer to peer mode.
7) Write a program using UDP Berkeley Sockets for wired/wireless network to enable file transfer (Script, Text, Audio and Video one file each) between two machines. Demonstrate the packets captured traces using Wireshark/Fiddler for traffic analysis tool in peer to peer mode.

Unit-VI Application Layer

8) Understanding protocol stack of Intranet
a) Analyze packet formats of Ethernet, IP, TCP and UDP captured using Wireshark/Fiddler for traffic analysis tool in peer to peer mode for wired and wireless networks.
b) Use any tool for custom packet generation (Packet Sender Tool) or write your own code for packet generation and analyze the packets.
9) Develop a client-server using C++ or JAVA to demonstrate the behavior of HTTP1.0, HTTP1.1, HTTP1.2 and HTTP2.0 protocols along with all success and error messages. Use Firefox as client browser.

List of Course Project areas:

1. Simulation of modulation and demodulation for digital telephone lines
2. Simulation of modulation and demodulation for 100 Mbps Ethernet Network
3. Simulation of modulation and demodulation for Gigabit Ethernet Network
4. Simulation of modulation and demodulation for 10Gigabit Ethernet Networks
5. Simulation of modulation and demodulation for 3G for mobile networks
6. Simulation of modulation and demodulation for 4G mobile networks
7. Develop a tool fox for line encoding methods
8. Develop a tool fox for modulation and demodulation methods
9. Design and deploy TCP based Multithreaded HTTP client server for accessing student activity data in the institute.
10. Design and deploy TCP based Multithreaded FTP client server to share institute level notices.
11. Design and deploy UDP based Multithreaded TFTP client server for your class

12. Design and deploy TCP based Multithreaded SMTP and POP3 mail client server for your campus.
13. Design and deploy TCP based Multithreaded Chat client server for your class.
14. Design and deploy UDP based Multithreaded Chat client server for your class.
15. Design and deploy UDP based Multithreaded Audio Conferencing client server for computer engineering department.
16. Design and deploy UDP based Multithreaded Video Conferencing client server for computer department
17. Implementation of RIP/OSPF/BGP using Packet Tracer
18. Simulation of AODV routing protocol using Packet Tracer/ NS3/OMNet ---------

List of Group Discussion Areas:

1. Energy-Efficient Architectures For Communication System
2. Satellite Communication System
3. Data Communication in Software Defined Networks
4. Cognitive Radios for Future Communication Frameworks
5. Fast Ethernet (Encoding Framing, Modulation, Multiplexing, Diameter etc)
6. Gigabit Ethernet (Encoding Framing, Modulation, Multiplexing, Diameter etc)
7. 10G Ethernet (Encoding Framing, Modulation, Multiplexing, Diameter etc)
8. IEEE 802.11b protocol based on HR-DSSS for wireless physical layer standard
9. IEEE 802.11g protocol based on ERP-OFDM for wireless physical layer standard
10. IEEE 802.11n protocol based on HT-OFDM for wireless physical layer standard
11. IEEE 802.11ac protocol based on VHT-OFDM for wireless physical layer standard

List of Home Assignment Areas:

Design:

1. Design a communication framework for irrigation system
2. Design a communication framework for automated car
3. Design a communication framework for smart city applications
4. RIP Routing Protocol for Intranet in VIT campus
5. OSPF Routing Protocol for Internet on India
6. BGP Routing Protocol for Asia continent

Case Study:

1. WiTricity technology for industrial applications
2. Multiple access schemes implemented in 4G mobile networks
3. RFCs for wired TCP based reliable communication
4. RFCs for wireless TCP based reliable communication
5. RFCs for SSL Certificates

Blog:

1. Journey of line encoding methods
2. Journey of modulation techniques
3. Internet Logical Addressing

4 Internet Routing Protocols

5. Applications Layer Protocols

Survey

1. Analogy to digital transformations on communication systems
2. Routing protocols for MANET
3. IEEE 802.1 Physical layer standard for Internet
4. IEEE 802.15.4 standard for IoT applications
5. IEEE 802.11 Wireless Standards for Wi-Fi

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
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. 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: ***(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

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 line encoding, modulation, topology, essential components of physical layer, data transmission rates to design computer networks.
2. Estimate reliability issues based on error control, flow control and pipelining by using bandwidth, latency, throughput and efficiency.
3. Propose mechanisms for server channel allocation in wired and wireless computer networks
4. Develop Client-Server architectures and prototypes by the means of correct standards, protocols and technologies
5. Analyze data flow between peer to peer in an IP network using Application, Transport and Network Layer Protocols
6. Compare sustainable engineering practice indicating the scientific purpose and utility of communication frameworks and standards.

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3	**2**			**2**									3		
2	3	2	**2**		2									3		
3	2	3	3	**2**		2	3							3		2
4	3	3	3	2	1	**2**	3	3	3				3	3	2	
5	3	3		2	3							**3**		3		
6	3	3		2						3		3		**3**		
Avg	**2.84**	**2.67**	**2.67**	**2.0**	**2**	**2**	**3**	**3**	**3**	**3**	**0**	**3**	**3**	**3**	**2**	**2**

Attainment Levels: 1, 5, 3, 4, 2, 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 – L5, CO3 – L3, CO4 – L4, CO5 – L2 and CO6 – L4

Future Course Mapping:

High Speed Networks, Wireless Networks, Mobile Networks, Network Security, Cyber Security

Job Mapping:

Network Engineer, Network Stack Developers, Application Developer

Syllabus Template FF No. :654

CS3205::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	PSO 2	PSO 3	PSO 4
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

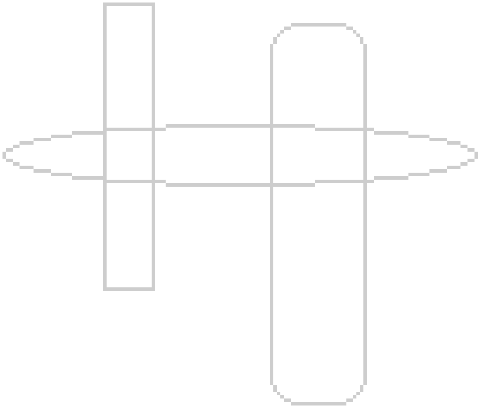
CO1 – L1, CO2 – L3, CO3 – L2, CO4 – L3, CO5 – L4 and CO6 – L5

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.



Syllabus Template **FF No. : 654**

CS3215::Web Technology

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

Tutorial: 1 Hours/Week

Lab: 2 Hours/Week

Course Prerequisites: Computer Networks

Course Objectives:

1. To describe most commonly used HTML5 and CSS3 tags and attributes for website development.
2. To associate event handling with HTML5 forms and CSS3 using JavaScript as a front-end technology for website development.
3. To extend HTML5 and CSS3 and JavaScript front end technologies with PHP and MySQL as a server side and backend technologies for website development.
4. To simplify website development using REST API and Spring Boot as server-side technologies.
5. To build single page applications using REACT as a reusable UI component technology as client-side technology
6. To assemble REACT as a front-end technology and Node JS as a server-side technology to develop enterprise applications

Course Relevance:

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

Section 1: Topics/Contents

Unit-I Front End Tools **04 Hours**

Introduction: Internet and WWW, Web site planning and design issues. HTML5: structure of html document, HTML elements: headings, paragraphs, line break, styles, colors, fonts, links, frames, lists, tables, images and forms, CSS, Bootstrap, XML, JSON.

Unit-II Client-Side Technologies 06 Hours

JavaScript: Overview of JavaScript, Data types, Control Structures, Arrays, Functions and Scopes, HTML5 forms Validation, Objects in JS, flex, DOM: DOM levels, DOM Objects and their properties and methods, Manipulating DOM. JQuery: Introduction to JQuery, Loading JQuery, selecting elements, changing styles, creating elements, appending elements, removing elements, handling events, Introduction to POSTMAN usages.

Unit-III Server-Side Technologies 04 Hours

PHP: Introduction to PHP, Features, sample code, PHP script working, PHP syntax, conditions & Loops, Functions, String manipulation, Arrays & Functions, Form handling, Cookies & Sessions, File Handling, Exception Handling, E-mail validations, MySQL with PHP, Laravel - The PHP Framework For Web Artisans. Introduction and usage of web3.0

Section2: Topics/Contents

Unit-IV Spring Boot 06 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-V React 04 Hours

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[4 Hrs]

Unit-VI Node JS 04 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.[4 Hrs]

List of Tutorials (13):

----- Django

1. Django - Environment, Creating a Project,
2. Apps Life Cycle, Admin Interface,
3. Creating Views, URL Mapping, Template System, Models,
4. Page Redirection, Sending E-mails,
5. Generic Views, Form Processing,
6. File Uploading, Apache Setup, Cookies Handling,
7. Sessions, Caching,

8. RSS
9. AJAX
10. -------- Introduction to JFrog Artifactory

    JFrog Artifactory is a DevOps solution for hosting, managing, and distributing binaries and artifacts. Any type of software in binary form – such as application installers, container images, libraries, configuration files, etc. – can be curated, secured, stored, and delivered using Artifactory. The name “Artifactory” reflects the fact that it can host any type of “artifact” needed in your software development “factory.”

List of Practical’s (Minimum Six):

1. Design and develop a responsive web page for your CV using multiple column layouts having video background. You can make the use of bootstrap as well as jQuery.

2. 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.

3. Design and develop a website to demonstrate (a) searching and sorting array for integer elements using JavaScript (b) array for named entities using JavaScript. You can make the use of bootstrap as well as jQuery.

4. Design and develop a responsive website to calculate Electricity bill using Django/Springboot/Node JS/ PHP. Condition for first 50 units – Rs. 3.50/unit, for next 100 units – Rs. 4.00/unit, for next 100 units – Rs. 5.20/unit and for units above 250 – Rs. 6.50/unit. You can make the use of bootstrap as well as jQuery.

5. Design and develop a responsive website to calculate Electricity bill using Django/Springboot/Node JS/ PHP. Condition for first 50 units – Rs. 3.50/unit, for next 100 units – Rs. 4.00/unit, for next 100 units – Rs. 5.20/unit and for units above 250 – Rs. 6.50/unit. You can make the use of bootstrap as well as jQuery.

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

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

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

9.Design and develop a responsive website for an online book store using REACT, Django/Springboot/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:
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 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. *Kumar, A., Web technologies, CRC press, 2019*
2. *Gupta, R., Internet & Web Technologies, Engineering Handbook, 2019*

3. *Martin, M.G., Programming for Beginners: 6 Books in 1 – Swift+PHP+Java+Javascript+Html+CSS: Basic Fundamental Guide for Beginners, independently published, 2018*
4. *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, O'Reilly Media; 5th edition, 2018*
5. *Kohli, S., Web Technologies, PPB Publications, 2015*
6. *Adam Bretz& Colin J Ihrig, "Full Stack Javascript Development with MEAN", SPD, First Edition 2015, Indian Reprint September 2015*
7. *GiulioZambon, "Beginning JSP, JSF and Tomcat", Apress Publication, Second Edition, 2013*
8. *Jeremy McPeak& Paul Wilton," Beginning JavaScript", Wrox Publication, Fifth Edition, 2015*
9. *Jeffrey C.Jackson, "Web Technologies: A Computer Science Perspective", Second Edition, Pearson Education, 2007, ISBN 978-0131856035.*
10. *Robert W. Sebesta: Programming the World Wide Web, 4th Edition, Pearson education, 2008*

Reference Books: *(As per IEEE format)*

1. *Marty Hall, Larry Brown,"Core Web Programming", Second Edition, Pearson Education, 2001, ISBN 978-0130897930.*
2. *H.M. Deitel, P.J. Deitel and A.B. Goldberg, "Internet & World Wide Web How To Program", Third Edition, Pearson Education, 2006, ISBN 978-0131752429.*
3. *Chris Bates: Web Programming Building Internet Applications, 3rd Edition, Wiley India, 2006.*
4. *Xue Bai et al: The web Warrior Guide to Web Programming, Thomson, 2003*

MOOCs Links and additional reading material:

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

Course Outcomes:

The student will be able to –

1. Create front end web pages using HTML5 and CSS3 tags and attributes
2. Provide validation mechanism and event handling in a website using JavaScript as a front end technology
3. Integrate front end with server side and backend technologies for commercial websites using PHP and MySQL
4. Write Web API/RESTful API application programming interface to communicate with Springboot as a server side technology.
5. Build single page applications using REACT as a reusable UI component technology as client side technology and Springboot and Node JS as server side technologies

6. Design and develop three tier enterprise application using client side, server side and back end technologies

CO-PO Map:

	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CS3215.1	2	2	2									3	1		2	
CS3215.2	2	2	2									3	1		2	
CS3215.3	3	3	3	1	2	2	2	2				3	1		3	2
CS3215.4	3	3	3	1	2	2	2	2				3	1		3	2
CS3215.5	3	3	3	1	2	2	2	2			2	3	1		3	2
CS3215.6	3	3	3	1	2	2	2	2	2	3	2	3	1		3	2
Average	**2.66**	**2.66**	**2.66**	**1.0**	**2**	**2**	**2**	**2**	**2.0**	**3.0**	**2.0**	**3.0**	**1.0**		**2.66**	**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 – L1, CO2 – L2, CO3 – L3, CO4 – L4, CO5 – L4 and CO6 – L5

Future Course Mapping:

Cloud Computing, Distributed System, Mobile Application Development

Job Mapping:

Software Engineer, Web Developer, IT Engineer, UI Developer

Syllabus Draft **FF No. : 654**

CS3061:: Software Modeling and Design (SMD)

Credits: 04 **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: 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. *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	PSO4
1	2	3	2									3	1		2	
2	2	3	2									3	1		2	
3	3	2	3		3	2	2	2			2	3	1	2	3	2
4	3	2	3		3	2	2	2	2		2	3	1	2	3	2
5	3	2	3		3	2	2	2			2	3	1	2	3	2
6	3	2	3		3	2	2	2		3	2	3	1	2	3	2
Average	**2.66**	**2.33**	**2.66**		**3**	**2**	**2**	**2**	**2.0**	**3.0**	**2.0**	**3.0**	**1.0**	**2.0**	**2.66**	**2.0**

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:

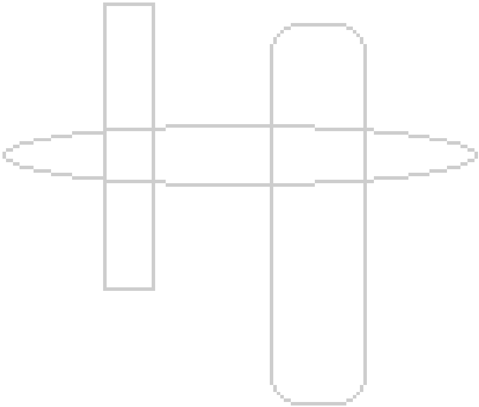
Software testing and Quality Assurance, Service-Oriented Software

Job Mapping:

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

T. Y. B. Tech. Computer Engineering
AY 2024-25

Module VI Course Content



Syllabus Template **FF No. : 654**

CS3226: 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			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CS3226.1	2	1			1									2	2	
CS3226.2	2	2	1	1	1									2	2	
CS3226.3	3	2	2	2	2		3	3						2	3	
CS3226.4	3	2	2	2	3	3			3					2	3	
CS3226.5	3	3	1	3	3				1		2			2	3	
CS3226.6	2	2	1	3	1					3		3			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	2.50	2.00

CO attainment levels:

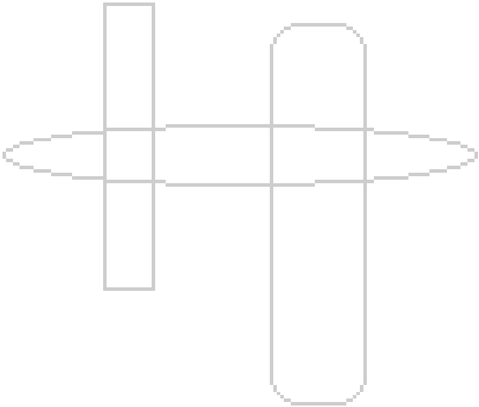
Attainment Levels:1,2,3,5,4,3

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



Syllabus Template **FF No. : 654**

CS3202: 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
2. Design of AI algorithm for Robot Navigation.
3. Design of AI algorithm for Customer Experience
4. Design of AI algorithm for Data-Informed Design
5. Design of AI algorithm for AI Decision Making
6. Design of AI algorithm for any application for Children
7. Design of AI algorithm for problems of Senior Citizens
8. Design of AI algorithm for ecommerce Applications
9. Design of AI algorithm for Enterprise UX Design
10. Design of AI algorithm as Teaching Aid for teachers

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

11. AI Trends
12. AI Research
13. AI Chatbot
14. Chatbot Magazine

15. AI Medical / Agriculture
16. AI Challenges
17. Knowledge based Inference Engine
18. Rule based inference Engine
19. Truth maintenance system
20. 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: 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
Laborarotory Practical: End Semester Examination: 100 Marks converted to 50Marks
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			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CS3226.1	2											2			2	
CS3226.2	2	2		1								2			2	
CS3226.3	2	3		2								2	2		3	
CS3226.4	2	3	3	2								2	2	2	3	
CS3226.5	2	3		2								2	2	2	3	
CS3226.6	2	3	3	2			3	2	2	2		3	3		3	2
Average	2.0	2.8	3.0	1.8			3.00	2.00	2.00	2.00		2.16	2.33	2.0	2.50	2.00

CO attainment levels:

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

Future Course Mapping:

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

Job Mapping:

What are the Job opportunities that one can get after learning this course
AI Data Analyst, Data Scientist

Syllabus Template **FF No. : 654**

CS3220:: Cyber Security

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.

Introduction to 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**

CS3053::Compiler Design

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

Course Prerequisites: Operating System, Theory of Computer Science

Course Objectives:

1. Understand the process of program execution cycle.
2. Understand the translation process from High Level Languages to Machine Level Language.
3. Know the syntax and semantic analysis approaches for efficient code/program verification.
4. Learn the methods of code generation which helps for the optimization.
5. Learn code optimization and runtime code synthesis.
6. Know the process of compiler design for emerging programming languages.

Course Relevance:

All high level programming languages are easy for users to understand but not understandable to a computing machine. The computing machine knows only binary data. A translation is required to convert higher level language into machine level, so that the intended program can be executed. This translation is done by using a compiler. This course will give you detailed insights of how compilers function internally. This gives freedom to design your own programming language with its compiler. The key technology of Compiler Design principles is to provide an in-depth view of translation and optimization process. Compiler design covers basic translation mechanism and error detection and recovery.

Section 1: Topics and Contents

Unit-I: Introduction to Compilers, interpreters, Assembler, Linker and Loader [04 Hours]

Compilers: Introduction to compiler phases, features of machine-dependent and independent compilers, overview of types of compilers, introduction to cross compiler, Interpreters: compiler vs. interpreter, phases, and working, Preprocessor: header file and macro expansion, Assembler: Introduction to Assembler, overview of types of Assembler, Linker and Loader: Introduction to Linker and Loader, overview of types of Linker and Loader.

Unit-II: Lexical Analysis and Syntax Analysis: [04 Hours]

Introduction to Compiler, Phases and Passes, Bootstrapping, Role of a Lexical Analyzer, Specification and Recognition of Tokens, LEX/FLEX, Expressing Syntax, Top-Down Parsing, Predictive Parsers, Implementing Scanners, Operator Precedence Parsers.

Unit-III: Syntax Analysis and Semantic Analysis: [05 Hours]

Bottom-Up Parsing, LR Parsers: Overview of types of LR Parsers, Constructing LALR parsing tables, Introduction to YACC/BISON, Type Checking, and Type Conversion, Symbol Table Structure.

Section 2: Topics and Contents

Unit-IV: Syntax-Directed Translation and Intermediate Code Generation: [05 Hours]

Syntax-Directed Definitions, Bottom-Up Evaluation, Intermediate Representations, and Intermediate Code Generation: various formats of intermediate codes, Error Detection & Recovery: Lexical Phase errors, syntactic phaseerrors, semantic errors. More about translation: Array references in arithmetic expressions.

Unit-V: Code Generation: [04 Hours]

Issues in Code Generation, Basic Blocks and Flow Graphs, Next-use information, Simple Code Generator, DAG representation of Basic Blocks, Peephole Optimization. Generating code from DAGs

Unit-VI: Code Optimization, Run-Time Environments and Data Flow Analysis: [06 Hours]

Code Optimization and Run-Time Environments: Principle Sources of Optimization, Optimization of basic blocks, Global Data Flow Analysis, Runtime Environments, and Source Language issues. Storage Organization, Storage Allocation strategies, Access to non-local names, Parameter Passing, Machine Dependent Optimization, Data Flow Analysis**:** Constant propagation, live range analysis.

Case studies: LLVM compiler Infrastructure, compiling OOP features, Compiling in multicore environment, Deep learning compilation, Parallel Compilers, Web Compilers.

List of Tutorials (13):

1. Single and two pass Assembler
2. Two pass Macro processor
3. Types of Linkers

4. Types of Loaders
5. Examples on First and Follow
6. Examples on Lex/Flex regular expressions
7. Construction of LL(1) parser
8. Construction of SLR parsing table
9. Construction of Canonical LR parsing table
10. Examples on YACC/Bison grammar rules
11. Translation Scheme
12. Examples of Intermediate code generation by Quadruples
13. Examples of DAG representation

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

1. LEX/FLEX specification and programming regular expressions.
2. Implement LEX/FLEX code to count the number of characters, words and lines in an input file.
3. Implement LEX/FLEX code to select only lines that begin or end with the letter 'a' and delete everything else.
4. Convert all uppercase characters to lowercase except inside comments.
5. Change all numbers from decimal to hexadecimal notation, printing a summary statistic (number of replacements) to stderr.
6. Implement Lexical Analyzer for language C.
7. Implement LR/SLR/LALR Parser.
8. YAAC specifications and implement Parser for specified grammar.
9. Implement Parser for language C.
10. Implement Syntax directed Translator.
11. Implement an Intermediate code generator (three address code and Quadruples)
12. Implement a code optimizer for C/C++ subset.
13. Implement a code generator for C/C++ subset.

List of Course Project areas:

1. Compiler for subset of C using Lex and YAAC.
2. Compiler for Subset of Java programming Language.
3. Intermediate Code generator.
4. Code Optimizer.

5. Develop an Editor for Assembly programming. (Use available Assembler MASM/TASM to compile the code and execute in editor).
6. Design a system to check syntax and semantics of English Language.
7. Design a system to check syntax and semantics of a subset of Logical programming Language.
8. Design a System to check syntax and semantics of a subset of Python programming language.
9. Compiler for subset of C++ programming language.
10. Compiler for a subset of Algol programming language.

List of Seminar Topics:

1. Tools complementary to Lex
2. Tools complementary to YAAC
3. Semantic Analyser
4. Obsolete programming Language compiler advantage and issues
5. Android App program compiler
6. Approaches of Intermediate Code generation
7. Recent Trends in Compiler
8. Recent Trends in Interpreter
9. Decompilation
10. Compilation in multicore machines

List of Design based Home Assignments:

1. Recent methodologies in Intermediate Code Generator
2. Recent methodologies in Code Optimizer
3. Universal Compiler
4. Compiler for Deep learning
5. Recent trend in parsers

---------

List of Case Study based Home Assignments:

1. Algol language Compiler
2. Compilation process (internals) of Functional Programming
3. Compilers for Mobile App development
4. LLVM compiler
5. Cross compiler

List of Blog based Home Assignments:

1. Decompilers: Ethical or Unethical?
2. Multi-paradigm programming compiler
3. State of the Art tools for rapid compiler development
4. Compiler for parallel machines
5. Compiler for distributed computing

List of Survey based Home Assignments:

1. Obsolete Programming Language Compilers
2. Obsolete Programming Language Interpreter
3. Compilers for various programming paradigms
4. Online compilers
5. Mobile app cross compiler

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
Home Assignment: 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. *Aho, A.V., Lam, M.S., Sethi, R., & Ullman, J.D. (2006). Compilers: Principles, Techniques,and Tools, Addison Wesley, ISBN 978-81317-2101-8 (2nd Edition).*

2. *Cooper, K., & Torczon, L. (2011). Engineering a compiler. Morgan Kaufmann, ISBN 155860-698-X.*

3. *Appel, A. W. (2004). Modern compiler implementation in C. Cambridge university press.*

4. *Appel, A. W., & Jens, P. (2002). Modern compiler implementation in Java. In ISBN 0–521–58388–8. Cambridge University Press.*
5. *Appel, A. W. (1998). Modern Compiler Implementation in ML, In ISBN 0-521-60764- .Cambridge University Press.*
6. *Raghavan, V. (2010). Principles of Compiler Design. Tata McGraw-Hill Education*

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

1. *Muchnick, S. (1997). Advanced compiler design implementation. Morgan Kaufmann, ISBN8178672413.*

2. *Levine, J. R., Mason, J., Levine,* J. R., Mason, T., Brown, D., Levine, J. R., & Levine, P.(1992). Lex *& yacc. "O'Reilly Media, Inc".*

MOOCs Links and additional reading material:

1. www.nptelvideos.in
2. https://swayam.gov.in/nd1_noc20_cs13/preview
3. https://www.udacity.com/course/compilers-theory-and-practice--ud168
4. https://online.stanford.edu/courses/soe-ycscs1-compilers

Course Outcomes:

The student will be able to –

1. Design basic components of a compiler including scanner, parser, and code generator.
2. Perform semantic analysis in a syntax-directed fashion using attributed definitions.
3. Apply local and global code optimization techniques.
4. Synthesize machine code for the runtime environment.
5. Develop software solutions for the problems related to compiler construction.
6. Adapt themselves to the emerging trends in language processing.

CO-PO Map:

	Program Outcomes (PO)												PSO			
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CO2	3	2	1	0	1	0	0	0	0	0	0	0	3	3	0	0
CO3	2	3	3	2	0	2	0	0	0	0	0	0	0	3	0	2
CO4	3	3	3	2	2	0	0	0	0	0	0	0	3	0	0	0
CO5	3	3	1	2	3	2	3	2	1	0	0	3	3	3	3	2
CO6	2	3	0	1	2	0	3	0	0	0	0	1	0	3	3	0
Average	**2.67**	**2.83**	**2**	**1.75**	**2**	**2**	**3**	**2**	**1**	**0**	**0**	**2**	**3**	**3**	**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

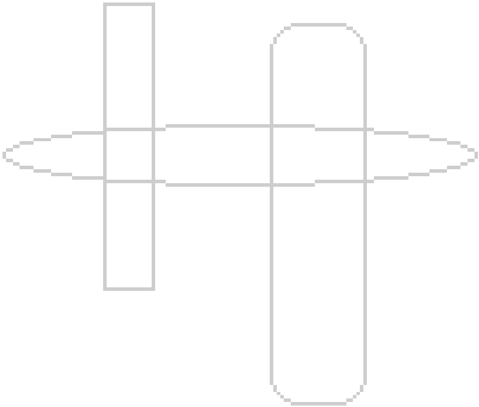
CO1 – L2, CO2 – L3, CO3 – L3, CO4 – L4, CO5 – L5 and CO6 – L5

Future Course Mapping: Parallel Compiler

Job Mapping: Compiler Engineer

B. Tech. Final Year Computer Engineering AY 2024-25

Module VII and VIII Course Content



Syllabus Template **FF No. : 654**

MD4202::Project Management

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

Course Prerequisites:

Production Systems, Basics of Industrial Engineering

Course Objectives:

1. Understand the Project Management Basics
2. Learn & Apply Project Management principles in Manufacturing and Service Sector
3. Learn Modern Tools like MS-Project for managing projects
4. Understand Project Management principles usage in the entrepreneurship

Section 1: Topics / Contents

Introduction:

Definition & Characteristics of Project, Performance Parameters: Time, Cost & Quality. Difference with respect to Standard Routine Production. Classification of Projects: Sector based, Investment based, Technology based, Causation based, Need based (BMERD) - Balancing, Modernization, Replacement, Expansion & Diversification Project Life Cycle Phases – Concept/Initiation Phase: Parameters Involved in Project Identification. Sources of New Project Ideas. Governmental Framework for Identification of Opportunities, Incentives from state & central govt.; Import-substitution projects

Project Conceptualization & Feasibility Analysis

Project Definition Phase: Project Formulation & Feasibility. Types of Feasibility Studies – Pre-feasibility, Support/Functional, Feasibility Study. Preparation of Project Feasibility Report

Project Planning, Implementation & Control

Planning & Organization Phase: Project Planning, Scheduling & Monitoring, Statement of Works, Project Specifications, Work Breakdown Structure, Network Analysis & Duration Estimating Network Diagrams – PERT/CPM, Estimate Activity Times, Milestone Scheduling. Project Crashing.

Section 2: Topics / Contents

Topics and Contents

Project Human Resource

Project Organization & Management. Project Organization Structure, Role of Project Manager, Resource Levelling, Resource Smoothening,

Project Cost Management

Project Cost Estimation: Need, Causes of Cost & Time Overruns. Nature of Cost Estimates, Types of Project Cost Estimates

Project Implementation & Control
Implementation Phase: Activities Involved: Erection & Commissioning, Installation, Trial Runs & Commencement of Commercial Production. Cleanup / Shutdown Phase

List of Home Assignments:
On following topics any type of (**Design, Case Study, Blog, Survey**) Home Assignment can be done in a group.

1. HA Based on Project Planning by using MS- Project
2. HA Based on CPM with or without using MS- Project (On actual projects)
3. HA Based on PERT with or without using MS- Project (On actual projects)
4. HA Based on Project Crashing (Cost Optimization)
5. HA Based on Resource Optimization - Resource Levelling and Smoothening
6. Survey based on PM Softwares and its applicability
7. Study based on Project Cost Management
8. Analysis based on Earned Value Management of any project
9. Blog based on Best practices in Project Management
10. Blog based on Project Management Softwares and its usage
11. Use of MS Project in Construction / IT/ Techonology/ NPD Projects
12. Use of Project Management in Instrumentation & Control Engg fields
13. Use of PM in Education (From student's perspective)
14. Feasibility study of any project or Business plan
15. Blog on Scope of Project Management domain in India
16. Blog on Challenges in Project Management for IT / Construction / Govt sector
17. Survey based on Project Manager's experience about Challenges/ Benefits in the PM domain
18. Study of Project Life Cycle Management
19. Study of Cost or Time or Quality aspects of Project Management
20. Study of Industry 4.O feasibility in Project Management

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.
MCQ Exam – Section I - Mid Semester 30 Marks converted to 30 equivalent Marks
Home Assignment - End of Semester 100 Marks converted to 10 equivalent Marks
MCQ Exam – Section II - End of Semester 30 Marks converted to 30 equivalent Marks
Comprehensive Viva Voce -End of Semester 100Marks converted to 30 equivalents Marks

Text Books: ***(As per IEEE format)***
1. Project Management- Harrold Kerzner, Wiley Publisher, 13th Edition, 2022
2. Project Management by Dennis Lock, 10th Edition, Gower Publishing Company, 2013
3. Project Management by Nagarajan, 2nd Edition, K, New Age International (P) Ltd Publishers, 2004

Reference Books: ***(As per IEEE format)***
1. Project Management - Body of Knowledge (BOK) Guide by PMI – Sixth Edition, 2021

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

Course Outcomes:

1. Learn the basic concepts of project and project management.
2. Ascertain the feasibility of small and medium projects with respect to managerial, marketing, operational, financial and socio-economic perspectives
3. Plan and schedule small and medium projects to achieve the triple constraint of time, cost and quality using software package
4. Understand the concept of earned value management system and critical chain in managing projects
5. Understand the concepts of project risk management and critical chain project management
6. Monitor the progress of projects to determine variances and recommend corrective actions

CO PO Map

	Program Outcomes (PO)												PSO			
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1									1	1	3	1				3
CO2									1	1	3	1				3
CO3	2				3				2	1	3	1		2		3
CO4		1									3	1				3
CO5		2									3	1				3
CO6						1	1				3	1				3
Average	**2.0**	**1.5**			**3.0**	**1.0**	**1.0**		**1.33**	**1.0**	**3.0**	**1.0**		**2.0**		**3.0**

CO Attainment Levels

CO-1 : 3
CO-2 : 3
CO-3 : 3
CO-4 : 3
CO-5 : 3
CO-6 : 3

Future Courses Mapping:

Industry Internship or Capstone Project

Job Mapping:

Project Manager, Project Executive, CFT member, Program Manager, Entrepreneurship

Syllabus Template **FF No. : 654**

LL4001: Generative AI

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

Course Prerequisites:
Statistical Mathematics, Artificial Intelligence

LinkedIn Course: For this course, each student will have to complete following six modular courses mentioned in six units. **Other guidelines related to examination and assessment will be given by course coordinator.**

Unit-I: Processing Text with Python Essential Training

In the world of big data, more and more information is consumed and analyzed in text form. Websites, social media, emails, and chats have become the key sources for data and insights. If you work with data, then understanding how to deal with unstructured text data is essential. In this course, instructor Kumaran Ponnambalam helps you build your text mining skill set, covering key techniques for extracting, cleansing, and processing text in Python. Kumaran reviews key text processing concepts like tokenization and stemming. He also looks at techniques for converting text into analytics-ready form, including n-grams and TF-IDF. Along the way, he provides examples of these techniques using Python and the NLTK library.

Unit-II: Hands-On Natural Language Processing

Dexterity at deriving insight from text data is a competitive edge for businesses and individual contributors. This course with instructor Wuraola Oyewusi is designed to help developers make sense of text data and increase their relevance. This is a hands-on course teaching practical application of major natural language processing tasks. Learn how to replicate the knowledge gained into the data that you work with. This course includes a background of each task’s process flow, use cases, and a coding demo. Some of the topics covered are named entity recognition, text summarization, topic modeling, and sentiment analysis.

Unit-III: Advanced NLP with Python for Machine Learning

An incredible amount of unstructured text data is generated every day by social media, web pages, and a variety of other sources. But without the ability to tame and harness that data, you'll be unable to glean any value from it. In this course, learn how to translate messy text data into powerful insights using Python. Instructor Derek Jedamski begins with a quick review of foundational NLP concepts, including how to clean text data and build a model on top of vectorized text. He then jumps into more complex topics such as word2vec, doc2vec, and recurrent neural networks. To wrap up the course, he lends these concepts a real-world context by applying them to a machine learning problem.

Unit-IV Deep Learning Foundations: Natural Language Processing with Tensor Flow

There is a growing demand to harness the power of natural language processing (NLP) and deep learning models to be able to make sense of textual data and reduce the emotional intervention of humans in order to make better decisions. In this course, instructor Harshit Tyagi provides a complete guide to understanding NLP using recurrent neural networks (RNNs). Harshit begins by introducing you to word encodings and using TensorFlow for tokenization. He describes the important concept of word embeddings and shows you how to use TensorFlow to classify movie reviews and project vectors. Harshit discusses RNNs and long short-term memory (LSTM), then shows you how to improve the movie review classifier from earlier in the course. He concludes with a discussion of how you can train RNNs to predict the next word in a sentence, which in turn allows you to generate some original text.

Unit-V Recurrent Neural Networks

Get started with recurrent neural network (RNN) concepts in a simplified way and build simple applications with RNNs and Keras. RNN is a fast-growing domain within the AI world. Popular groundbreaking applications like language translation, speech synthesis, question answering, and text generation use RNNs as their base technology. Studying this technology, however, has several challenges. Most learning resources are math heavy and are difficult to navigate without good math skills. IT professionals from varying backgrounds need a simplified resource to learn the concepts and build models quickly. In this course, Kumaran Ponnambalam provides a simplified path to studying the basics of recurrent neural networks, allowing you to become productive quickly. Kumaran starts with a simplified introduction of RNN before walking through the process of building a model. He then covers the popular building blocks of RNN with GRUs, LSTMs, word embeddings, and transformers.

Unit VI Generative AI: Working with Large Language Models

Transformers have quickly become the go-to architecture for natural language processing (NLP). As a result, knowing how to use them is now a business-critical skill in your AI toolbox. In this course, instructor Jonathan Fernandes walks you through many of the key large language models developed since GPT-3. He presents a high-level overview of GLaM, Megatron-Turing NLG, Gopher, Chinchilla, PaLM, OPT, and BLOOM, relaying some of the most important insights from each model.Get a high-level overview of large language models, where and how they are used in production, and why they are so important to NLP. Additionally, discover the basics of transfer learning and transformer training to optimize your AI models as you go. By the end of this course, you’ll be up to speed with what’s happened since OpenAI first released GPT-3 as well as the key contributions of each of these large language models.

Syllabus Template **FF No. : 654**

CS4217:: Human-Computer Interaction (HCI)

Credits: 02 **Teaching Scheme Theory: 2 Hours/Week**

Course Prerequisites: NA

Course Objectives:

1. To categorize IT applications based on measurable human factors,
2. To study the user community through user survey and/or field visit,
3. To design user-friendly user interfaces with due consideration of interface theory and principles,
4. To apply usability evaluation methods to identify the usability issues with IT applications,
5. To understand the kind of documentation required for IT applications,
6. To integrate web and mobile app design approaches as per user requirements.

Course Relevance:

Human-Computer Interaction (HCI) is a multi-disciplinary socio-technical course, with a goal of bringing the power of computers and communication systems to users, customers or people. It aims to make all computing and communications systems more accessible, maintainable and useful in working, learning and recreational life of their users. It helps every computing, web or mobile application to become really user-centered, increasing its number of users as well as related sales.

Section 1: Topics / Contents

Unit-I Fundamentals of Human Computer Interaction: **05 Hours**

Definition of HCI, Interdisciplinary Nature, Related Disciplines, Usability, Types of Usability, User Interface (UI), Measurable Human Factors, Accessibility, Differently-abled Users.

Unit-II Interaction Concepts and Models: **05 Hours**

User Persona, User Categorization, Golden Rules of Interface Design, Miller's Principle, Task Analysis - GOMS, Contextual Inquiry, Work Models, Interaction Styles.

Unit-III Design Process: **04 Hours**

Design Concept, Three Pillars of Design, Process of Design, Ethnographic Observations, Participatory Design, Internationalization.

Section 2: Topics/Contents

Unit-IV Usability Evaluation: **05 Hours**

Expert-based Evaluation, User-based Evaluation, Formative Evaluation, Summative Evaluation, Heuristic Evaluation, Cognitive Walkthrough, Semiotic Analysis, Icon Categorization, User Surveys, Interviews, Usability Testing.

Unit-V Documentation and Groupware: 05 Hours
Classification of Documents, Reading from Displays, Online Help, Tutorials, Error / Warning Messages, Groupware, Computer Supported Cooperative Work (CSCW), Dimensions of Cooperation, Challenges with Online Communications.

Unit-VI Website and Mobile App Design: 04 Hours
Content Design, Interaction and Navigation Design, Presentation Design, Differences in Design Approaches, Design and Evaluation Tools.

List of Home Assignments:
Design:

6. Social Network for Spiritual Users
7. App for Alzheimer’s disease
8. Health Tracking App
9. Ration Card Management App
10. Innovative e-Commerce Platform

Case Study:

6. Chatbot in Healthcare Domain
7. Best Food Ordering App in India
8. Online Teaching-Learning Process
9. Use of Twitter with Indian Users
10. User Experience with Car Booking Apps

Blog

6. Noise of Notifications
7. Challenges in Food Delivery Service
8. Need for Accessibility Guidelines
9. Usability of Autonomous Vehicles
10. Failure of Usability Testing

Surveys

7. User Experience with Video-Conferencing Apps
8. User Errors on Social Networking Sites (SNS)
9. Challenges for Hearing Impaired Users with IT Applications
10. Most Popular Indian Mobile Apps (Made In/By India)

11. Impact of Ban on Chinese Apps in India

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.
MCQ Exam – Section I - Mid Semester 30 Marks converted to 30 equivalent Marks
Home Assignment - End of Semester 100 Marks converted to 10 equivalent Marks
MCQ Exam – Section II - End of Semester 30 Marks converted to 30 equivalent Marks
Comprehensive Viva Voce -End of Semester 100Marks converted to 30 equivalents Marks

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

1. Ben Shneiderman, "Designing the User Interface", Third Edition, Pearson Education, ISBN 81-7808-262-4.
2. Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale, "Human-Computer Interaction", Third Edition, Pearson Education, ISBN 81- 297-0409-9.

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

1. Donald Norman, "The Design of Everyday Things", 2002 Edition, Basic Books, ISBN 100-465-06710-7.
2. Wilbert Galitz, "The Essential Guide to User Interface Design", Second Edition, Wiley-Dreamtech India (P) Ltd., ISBN 81-265-0280-0.
3. John Carroll, "Human-Computer Interaction in the New Millennium", Pearson Education, ISBN 81-7808- 549-6.

MOOCs Links and additional reading material:

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

Course Outcomes: Students will be able to

1. Appreciate the differences among IT applications and their categories based on measurable human factors.
2 Study the user community through user survey and/or field visit.
3. Design user-friendly user interfaces as per user requirements and UI design principles.
4. Apply a suitable usability evaluation method to identify the usability issues.
5. Understand the kind of documentation required for IT applications.
6. Enhance UI designs as per desired web or mobile app design approach.

	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO 8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CS4217.1	2	3	-	-	-	-	-	-	-	-	-	2	-	-	-	2
CS4217.2	2	3	-	2	-	-	-	-	-	-	-	2	-	-	-	2
CS4217.3	3	2	3	-	2	3	2	2	2	-	-	2	-	-	3	2
CS4217.4	2	3	3	2	-	-	-		-	-	-	2	-	-	2	2
CS4217.5	3	-	-	-	-	-	-	-	-	2	1	-	-	-	2	2
CS4217.6	3	2	3	2	2	3	2	2	2	-	-	-	-	-	3	2
Average	**2.5**	**2.6**	**3.0**	**2.0**	**2.0**	**3.0**	**2.0**	**2.0**	**2.0**	**2.0**	**1.0**	**2.0**	**-**	**-**	**2.5**	**2.0**

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

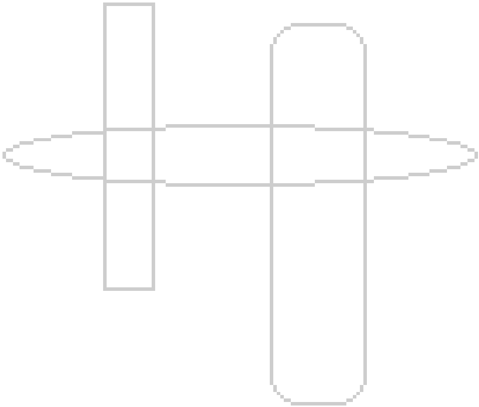
CO1 – L3, CO2 – L3, CO3 – L2, CO4 – L2, CO5 – L1 and CO6 – L3

Future Course Mapping:

User Interface Design, Usable Security & Intelligent User Interfaces

Job Mapping:

UI Designer, Product Designer, Software Engineer, Mobile App Developer and Other Jobs



Syllabus Template **FF No. : 654**

CS4272:: Neural Networks

Credits: 02 **Teaching Scheme Theory: 2 Hours/Week**

Course Prerequisites: Fundamentals of Mathematics, Python Programming.

Course Objectives: In this course, we will study the following topics

1. To Study the neural network models: multilayer perceptron, distance or similarity Study the based neural networks, associative memory and self-organizing feature map, radial basis function, multilayer perceptron.
2. To understand the basic neural network architectures and learning algorithms, for applications in pattern recognition, image processing, and computer vision.
3. To Study the delta learning rule, the backpropagation delta learning rule, self-organization learning, and the r4-rule.
4. To Understand the importance of tolerance of imprecision and uncertainty for the design of robust and low-cost intelligent machines.
5. To apply basic principles of Artificial Neural networks in solutions that requires problem-solving, inference, perception, knowledge representation, and learning.
6. To build Neural Network models and implement them in real-life scenarios for different applications like detection and classification.

Course Relevance: Neural networks reflect the behavior of the human brain, allowing computer programs to recognize patterns and solve common problems in the fields of AI, machine learning, and deep learning. Artificial neural networks (ANNs) are comprised of node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

SECTION-I: Topics and Contents:

Unit-I **Introduction:** Biological Neuron – Artificial Neural Model - Types of activation functions – Mc_Culloch_Pitts, Hebnet, Architecture: Feedforward and Feedback, Convex Sets, Convex Hull and Linear Separability, Non-Linear Separable Problem. XOR Problem, Multilayer Networks, Backpropagation algorithm

Unit-II **Learning Techniques:** Supervised, Unsupervised, Reinforcement Learning. Learning Algorithms, Error correction and Gradient Descent Rules, Learning objective of TLNs, Perceptron Learning Algorithm, Perceptron Convergence Theorem

Unit-III **Supervised Learning:** Perceptron learning and Non Separable sets, α-Least Mean Square Learning, MSE Error surface, Steepest Descent Search, μ-LMS approximate to gradient descent, Application of LMS to Noise Cancelling, Genetic Algorithm, Regression analysis

SECTION-II :

Topics and Contents:

Unit-IV S**upport Vector Machines and Radial Basis Function:** Learning from Examples, Statistical Learning Theory, Support Vector Machines, SVM application to Image Classification, Radial Basis Function Regularization theory, Generalized RBF Networks, Learning in RBFNs, RBF application to face recognition

Unit-V **Attractor Neural Networks:** Associative Learning Attractor Associative Memory, Linear Associative memory, Hopfield Network, application of Hopfield Network, Brain State in a Box neural Network, Simulated Annealing, Boltzmann Machine, Bidirectional Associative Memory

Unit-VI **Self-organization Feature Map**: Maximal Eigenvector Filtering, Extracting Principal Components, Generalized Learning Laws, Vector Quantization, Self-organization Feature Maps, Application of SOM, Growing Neural Gas.

List of Home Assignments:

List of Design Based Home Assignments

HA_D1.Based on Integrate-and-Fire Neurons
HA_D 2.   Based on Mac_Culloch_PittsNeuronsmodel
HA_D 3. Based on Heb Net
HA_D 4. Based on the Backpropagation algorithm.
HA_D 5. Based on Support Vector Machine
HA_D 6. Based on Linear and nonlinear models.
HA_D 7. Based on K means Clustering algorithm.
HA_D 8. Based on Boltzmann Machine.
HA_D 9. Based on a self-organizing feature map.
HA_D 10. Based on Integrate-and-Fire Neurons

List of Case Study Based Home Assignments

HA_CS 01 A case study of using artificial neural networks for classifying causes of death from verbal autopsy.

HA_CS 02A case study of using artificial neural networks for wind energy

HA_CS 03A case study of using artificial neural networks for medical diagnostics.

HA_CS 04A case study of using artificial neural networks for the banking sector.

HA_CS 05A case study of using artificial neural networks for Industry.

HA_CS 06A case study of using artificial neural networks forthe Film industry.

HA_CS 07A case study of using artificial neural networks forthe agriculture sector.

HA_CS 08A case study of using artificial neural networks for Education system or teaching-learning process.

HA_CS 09A case study of using artificial neural networks forsafety and security.

HA_CS 10A case study of using artificial neural networks for rainfall prediction.]

List of Blog Based Home Assignment

HA_Blog 01:Two Dimensional Feature Extraction and Blog Classification using Artificial Neural Network

HA_Blog 02:An intelligent personalized web blog searching technique using neural networks

HA_Blog 03:An author gender detection method using whale optimization algorithm and artificial neural network

HA_Blog 04:A comparative study of machine learning techniques in blog comments spam filtering

HA_Blog 05:A document-level sentiment analysis approach using artificial neural network and sentiment lexicons

HA_Blog 06:Automatic classification of unstructured text

HA_Blog 07:Product related information sentiment-content analysis based on neural networks

HA_Blog 08:Customer segmentation and classification by using data mining: an example of VOIP phone

HA_Blog 09:Travel blogs and the implications for destination marketing

HA_Blog10:An event driven neural network system for evaluating public moods from online users' comments

List of Survey Based Home Assignments

HA_Survey 01: State-of-the-art in artificial neural network applications: A survey

HA_Survey 02: -Personality traits analysis using Artificial Neural Networks: A survey

HA_Survey 03: Artificial neural networks in accounting and finance: Modeling issues

HA_Survey 04: Neural networks for control systems—a survey

HA_Survey 05: Student's success prediction model based on artificial neural networks (ANN) and a combination of feature selection methods

HA_Survey 06: Reusability in artificial neural networks: An empirical study

HA_Survey 07: Survey on neural networks used for medical image processing

HA_Survey 08: Artificial neural networks: A powerful tool for cognitive science, A Survey

HA_Survey 09: A model selection approach to real-time macroeconomic forecasting using linear models and artificial neural networks

HA_Survey10: Predicting students' performance based on learning style by using artificial neural networks

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.
MCQ Exam – Section I - Mid Semester 30 Marks converted to 30 equivalent Marks
Home Assignment - End of Semester 100 Marks converted to 10 equivalent Marks
MCQ Exam – Section II - End of Semester 30 Marks converted to 30 equivalent Marks
Comprehensive Viva Voce -End of Semester 100Marks converted to 30 equivalents Marks

Text Books: *(As per IEEE format)*

1. *Introduction to Artificial Neural Systems Jacek M. Zurada, JAICO Publishing House Ed. 2006.*
2. *Neural Networks A Classroom Approach —Satish Kumar, McGraw Hill Education (India) Pvt. Ltd, Second Edition.*
3. *Neural Networks a Comprehensive Foundations, Simon S Haykin, PHI Ed.*

Reference Books: *(As per IEEE format)*

1. *Neural Networks A Classroom Approach —Satish Kumar, McGraw Hill Education (India) Pvt. Ltd, Second Edition.*
2. *Neural Networks in Computer Inteligance, Li Min Fu TMH 2003*
3. *Neural Networks -James A Freeman David M S Kapura Pearson Ed., 2004.*
4. *Artificial Neural Networks – B. Vegnanarayana Prentice Hall of India P Ltd 2005*

MOOCs Links and additional reading material:

1. www.nptelvideos.com
2. www.coursera.com

Course Outcomes: On the completion of course, student will able to

1. Understand the concepts, techniques, and building blocks of neural network models like Mac_Culloch_Pitts, Hebnet and Artificial Neural Networks.
2. distinguish between supervised, unsupervised and reinforcement learning techniques.
3. Apply basic principles of Artificial Neural networks in solutions that require problem-solving, inference, perception, knowledge representation, and learning.
4. Implement neural network models to solve real-world problems.
5. Evaluate different most appropriate parameters for performance calculation.
6. Build Neural Network models and implement them in real-life scenarios for different applications.

Syllabus Template **FF No.: 654**

CS4222:Image Processing

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

Course Prerequisites: Digital logic Design, Microprocessor, Computer Organization.

Course Objectives:

1.To describe different color models and image processing techniques.
2.To analyze image condition and deduce enhancement algorithms.
3.To apply image segmentation to identify the region of interest
4.To develop an algorithm to recognize the specified objects in the given image.
5.To study different image morphological operation.
6.To learn different image compression techniques.

Course Relevance:

Vision sense is the most powerful human sense organ. In the world where intelligent automation is taking place, image processing is a vital domain for research and development. In Industry 4.0, image processing systems built around industrial cameras are an essential component in automated production. Throughout all steps of production, from the inspection of raw materials and production monitoring (i.e., flaw detection) to final inspections and quality assurance, they are an indispensable part of achieving high efficiency and quality standards. In the Entertainment Industry, latest trends such as 4K video streaming requires high quality compression that can provide limited/no loss image quality with high fps. In social networking, sharing images has been a vital part. Creating innovative effects and overall manipulating the images will be explored

Section 1: Topics/Contents

Unit-I Introduction **4 Hours**

Elements of image processing system, Scenes and Images, Vector Algebra, Human Visual System, color vision color model: RGB, HVS, YUV, CMYK, YCbCr and some basic relationships between pixels, linear and nonlinear operations, Image sampling and quantization.

Unit-II Image Enhancements **5 Hours**

Memory-less operations, Spatial domain image enhancements: Denoising filters, Smoothing Operation, Sharpening Operation, and Contrast stretching /enhancement, histogram and histogram equalization.

Unit-III Image segmentation **5 Hours**

Classification of image segmentation techniques: Edge-based Segmentation, Region based techniques. Binarization: Global Thresholding, Adaptive thresholding. Types of Edge detector: derivative filters, Sobel, Canny. Edge linking. Feature Extraction.

Section2:Topics/Contents

Unit-IV Morphological Operation 4 Hours

Binary Morphology, Erosion Dilation, Opening and Closing.

Unit-V Feature Extraction and Object Recognition 5 Hours

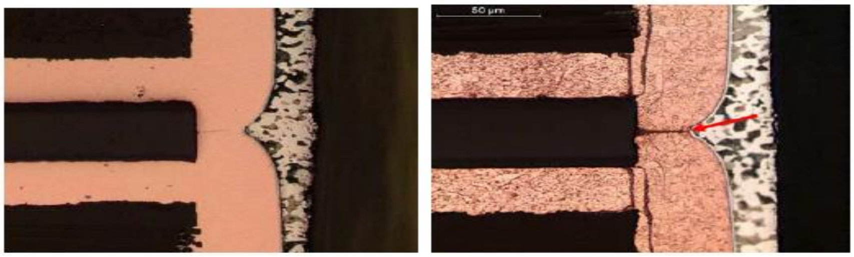
Feature points and feature detection (Line, circle and corner). Line detection: RANSAC, Hough Transform. Corner detection: Harris Corner Detector. Feature descriptors, Descriptor matching. SIFT, Boundary representation (Chain code), Boundary detection-based techniques.

Unit-VI Image Compression 5 Hours

Introduction and need, Coding redundancy, classification of compression techniques (Lossy and lossless- JPEG, Run Length Coding, Huffman Coding, Shannon Fano coding).

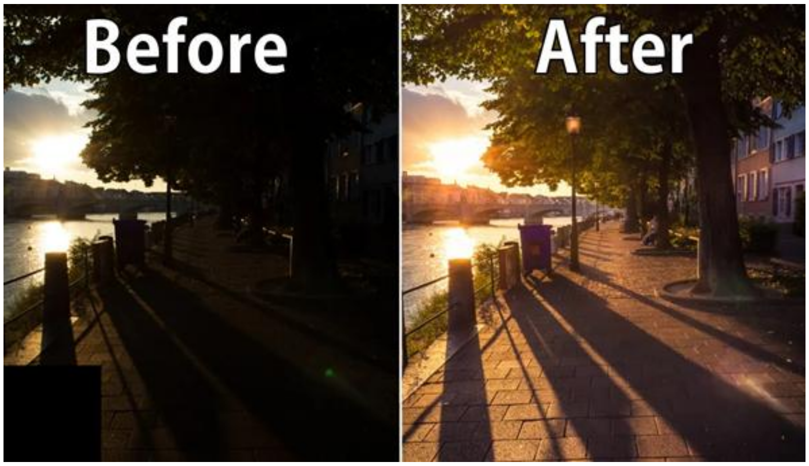
List of Design based Home Assignments:

1.Design an algorithm to identify fault in a "PCB inspection system" as shown below

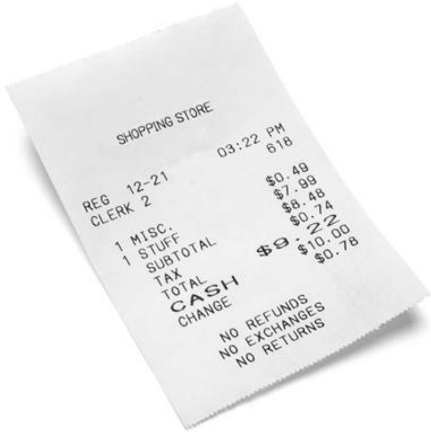


2.Design an algorithm to perform segmentation of the image below to extract the mango from its background.

3.Design an algorithm to get from image 1 to image 2.

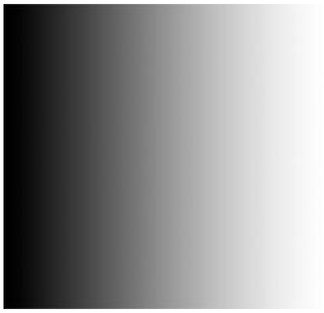


4.Design an algorithm to recognize character “0” in the image below.



SHOPPING STORE
03:22 PM
618
REG 12-21
CLERK 2
1 MISC. $0.49
1 STUFF $7.99
SUBTOTAL $8.48
TAX $0.74
TOTAL $9.22
CASH $10.00
CHANGE $0.78
NO REFUNDS
NO EXCHANGES
NO RETURNS

5.Design an algorithm to compress a 300x300 pixel image with horizontal black to white gradient as shown below.



List of Case Study based Home Assignments:

1. Cam-scanner: Document scanning app
2. Tesseract OCR library
3. Instagram filters
4. OpenCV
5. Google Street View

List of Blog based Home Assignments:

1. Image processing on Embedded platforms
2. Face recognition system security analysis for authentication
3. Image processing in MSME for effective automation
4. H.264 codec for image streaming
5. Role of mathematics in image processing

List of Survey based Home Assignments:

1. Image quality metrics
2. Vision based self-driving car safety
3. Compression techniques & codecs
4. State of the art applications such as AR/ XR
5. Human recognition in social networking apps like Facebook

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.
MCQ Exam – Section I - Mid Semester 30 Marks converted to 30 equivalent Marks
Home Assignment - End of Semester 100 Marks converted to 10 equivalent Marks
MCQ Exam – Section II - End of Semester 30 Marks converted to 30 equivalent Marks
Comprehensive Viva Voce -End of Semester 100Marks converted to 30 equivalents Marks

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

1. *Rafael Gonzalez & Richard Woods, "Digital Image Processing," 3rd Edition, Pearson publications, ISBN 0132345633.*

2. *Anil K. Jain, "Fundamental of Digital Image Processing," 5th Edition, PHI publication, ISBN 13: 9780133361650.*

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

1.*Pratt, "Digital Image Processing," Wiley Publication, 3rdEdition, ISBN 0-471- 37407-5.*

2. *K.R. Castleman, "Digital Image Processing,"3rd Edition, Prentice Hall: Upper Saddle River, NJ, 3, ISBN 0-13-211467 -4.*

MOOCs Links and additional reading material:

1. https://nptel.ac.in/courses/117/105/117105135/
2. https://nptel.ac.in/courses/106/105/106105032/
3. https://nptel.ac.in/noc/courses/noc19/SEM2/noc19-cs58/
4. https://www.coursera.org/learn/computer-vision-basics

Course Outcomes:

The student will be able to –

1.Recognize different color models and image processing techniques. (1)

2.Select image enhancement algorithm to improve the quality of image. (2)

3.Build image segmentation techniques to identify region of interest. (4)
4.Predict image morphological techniques to resize the image. (3)
5.Construct an algorithm to recognize the specified objects in the given image. (5)
6. Identify different image compression techniques to reduce the size of image. (3)

CO-PO Map:

	Program Outcomes (PO)												PSO			
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	3	2		1	-	-	-	-	-	-	-	2	2	2	3	3
CO2	3	2	2	1	3	-	-	2	-	-	-	2	2	2	3	3
CO3	2	2	3	2	3	-	-	-	1	-	-	2	2	2	3	3
CO4	3	3	3	2	3	2	-	-	-	-	-	2	2	2	3	3
CO5	3	2		2	3	2	-	-	1	1	2	2	2	2	3	3
CO6	3	2		2	3	2	2	2		1	2	2	2	2	3	3
Average	**2.8**	**2.16**	**2.7**	**1.88**	**3**	**2.0**	**2.0**	**2.0**	**1.0**	**1.0**	**2.0**	**2.0**	**2.0**	**2.0**	3.0	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 – L2, CO2 – L2, CO3 – L4, CO4 – L3, CO5 – L5 and CO6 – L3

Future Course Mapping:

1. Augmented Reality
2. Multimedia Processing

Job Mapping:

1. Augmented Reality Experience Designer
2. Automation Engineer
3. Embedded Software Developer
4. Image Processing Expert

Syllabus Template **FF No.: 654**

ET4230 : Natural Language Processing

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

Course Prerequisites: Theory of Computer Science, Compiler Design

Course Objectives:

1. To understand morphology for given natural language
2. To learn how to design lexical analyzer for given natural language
3. To learn how to design Syntactic Analyzer for given natural language
4. To learn how to design type dependency parser using pragmatic approach for given natural language
5. To understand the scientific process for machine transliteration, machine translation and information retrieval for given natural language using statistical approach

Course Relevance: Although Natural Language Processing (NLP) has been with us for quite some time, it has only recently gained industry-wide attention, thanks to Deep Learning. Today, NLP is a core competence area in Data Science and IT, with applications spanning across sectors that rely on harnessing language data's potential. Essentially, NLP applications are designed to extract relevant and meaningful information from natural human language data and impart machines with the ability to interact with humans.

SECTION 1: TOPICS/CONTENTS

Unit-I Introduction [5 Hours]

What is natural language processing? Applications of NLP, Origins of NLP, Challenges of NLP, Language and Knowledge, Language and Grammar, Processing Indian Languages. Grammar-based language models, lexical functional Grammar(LFG), Government and Binding (GB), Lexical functional Grammar Model, Generative grammars, Statistical Language Model.

Unit-II Regular Expressions and Automata [5 Hours]

Formal Language Theory: Basic Notions, Basic Regular Expression Patterns, Disjunction, Grouping and Precedence, Advanced Operators, Substitution, Finite State Automata, NFSA. Words and Transducers, Morphology, Inflectional Morphology, Derivational Morphology, Finite State

Morphological Parsing, Construction of Finite State Lexicon, Finite State Transducers, FST for Morphological Parsing.

Unit-III Theory of parsing / Syntactic Analysis [4 Hours]

Context Free Grammar, parsing, Top-down Parsing, Bottom-up parsing, Probabilistic parsing, Indian Languages parsing Semantic Analysis: Meaning Representation, Lexical Semantic, Ambiguity, Word Sense Disambiguation, Discourse processing, Natural Language Generation.

SECTION2: TOPICS/CONTENTS

Unit-IV Computer Linguistics [5 Hours]

Machine Transliteration using Statistical Language modeling: N-gram model, Machine Transliteration: Rule-based, Phonology and Stress Analysis based and Statistical based, Support vector machine, Memory Entropy Model, Hidden Markov Model, Conditional Random Fields, Evaluation Metrics

Unit-V Machine Translation [5 Hours]

Introduction, Problems in MT, Characteristics of Indian Languages, Machine Translation Approaches, Direct Machine Translation, Rule-Based MT, Corpus Based Machine Translation, Semantic/Knowledge based MT Systems, Translation involving Indian Languages, Statistical-Based using MT Tools - GIZA++, SRTLM and Moses, Evaluation Metrics

.

Unit-VI Information Retrieval [4 Hours]

Designing features for IR Systems, IR Models, Classical IR Models, Non Classical IR Models. Evaluation of IR Systems, NLP in IR, Relation Mapping, and Knowledge based Approaches, Conceptual Graphs in IR, Cross Language Information Retrieval, Evaluation Metrics.

List of Course Project Areas:

1. Biomedical Text Mining.
2. Computer Vision and also NLP.
3. Deep Linguistic Processing.
4. Controlled Natural Language.
5. Language Resources and also Architectures for NLP.
6. Sentiment Analysis and also Opinion Mining.
7. Recognizing Similar Texts
8. Inappropriate Comments Scanner
9. Language Identifier
10. Image-Caption Generator

List of Design based Home Assignments:

Design:

1. Use a simple method to classify positive or negative sentiment in tweets
2. Use a more advanced model for sentiment analysis
3. Use vector space models to discover relationships between words and use principal component analysis (PCA) to reduce the dimensionality of the vector space and visualize those relationships
4. Write a simple English-to-French translation algorithm using pre-computed word embeddings and locality sensitive hashing to relate words via approximate k-nearest neighbors search
5. Create a simple auto-correct algorithm using minimum edit distance and dynamic programming
6. Write a better auto-complete algorithm using an N-gram model (similar models are used for translation, determining the author of a text, and speech recognition)
7. Write your own Word2Vec model that uses a neural network to compute word embeddings using a continuous bag-of-words model

6. Train a neural network with GLoVe word embeddings to perform sentiment analysis of tweets

8. Train a recurrent neural network to perform NER using LSTMs with linear layers

Translate complete English sentences into French using an encoder/decoder attention model

9. Build a transformer model to summarize text

1. Build a chatbot using a reformer model.

Case Study:

1. Clinical Documentation
2. Speech Recognition
3. Computer-Assisted Coding (CAC)
4. Data Mining Research
5. Automated Registry Reporting
6. Clinical Decision Support
7. Clinical Trial Matching
8. Prior Authorization

Blog:

1. Machine Translation: Rule-Based
2. Machine Translation: Statistical-Based
3. MT Tools - GIZA++, SRTLM and Moses
4. GIZA++, SRTLM and Moses
5. Natural Language Resources for Beginners
6. Natural Language Resources for Practitioners
7. Biomedical Text Mining
8. Computer Vision and also NLP
9. Deep Linguistic Processing
10. Controlled Natural Language.

11. Language Resources and Architectures for NLP
12. Sentiment Analysis and also Opinion Mining
13. NLP includes Artificial Intelligence

Survey:

1. Language Models
2. Top-down Parsing
3. Bottom-up parsing
4. Probabilistic parsing
5. Indian Languages parsing

Assessment Scheme: Ensures 360 degree assessment and covers all aspects of Bloom's Taxonomy.
MCQ Exam – Section I - Mid Semester 30 Marks converted to 30 equivalent Marks
Home Assignment - End of Semester 100 Marks converted to 10 equivalent Marks
MCQ Exam – Section II - End of Semester 30 Marks converted to 30 equivalent Marks
Comprehensive Viva Voce -End of Semester 100Marks converted to 30 equivalents Marks

Text Books:

1.Tanveer Siddiqui and U S Tiwary, "Natural Language Processing and Information Retrieval" Fourth Impression, Oxford, ISBN-13:978-019-569232-7.

2. Daniel Jurafsky and James H Martin., "Speech and Language Processing", 2nd edition, Pearson, Second Impression-2014,ISBN: 978-93-325-1841-43.

3. Christopher D.Manning and Hinrich Schutze,, "Foundations of Statistical Natural Language Processing" , MIT Press, 1999.

Reference Books:

1. *Alexander Clark (Editor), Chris Fox (Editor), Shalom Lappin (Editor), "The Handbook of Computational Linguistics and Natural Language Processing".*
2. *Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit, Steven Bird, Ewan Klein, and Edward Loper, O'Reilly Publication*
3. *Natural Language Processing with Python CookBook, Krishna Bhavsar, Naresh Kumar, Pratap Dangeti, Packt Publication.*
4. *Ralph Grishman, " Computational Linguistics: An Introduction (Studies in Natural Language Processing)", Cambridge University Press, ASIN : B01MQYCTOB.*

MOOCs Links and additional reading material:

- www.nptelvideos.in
- www.nfnlp.com
- https://www.mooc-list.com/tags/nlp
- https://www.my-mooc.com/en/mooc/natural-language-processing-nlp/
- https://huggingface.co/learn/nlp-course/
- https://www.coursera.org/learn/attention-models-in-nlp

Course Outcomes:

The student will be able to:

1. Interpret morphology for given natural language (2)
2. Construct shallow depth lexical analyzer and syntactic analyzer for given natural language(3)

3. Develop shallow depth type dependency parser for given natural language(3)
4. Develop shallow depth machine transliteration, machine translation, information retrieval for given natural language using either linguistic or statistical approach (3)
5. Correlate shallow depth machine transliteration, machine translation, information for given natural language (4)
6. Evaluate machine transliteration, machine translation and information retrieval results using standard evaluation metrics (5)

CO-PO Map:

CO	Program Outcomes (PO)												PSO			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3	PSO 4
1	3	2											2			
2	3	2	3	2									3	3		
3	3	2	3	2										3	3	
4	3	2	3	3	3				1					3	3	
5	3	3	3	3	3	2			1		3			2	3	
6	3	2	3	3	3	2						2		2	3	3
Avg	3	2.16	3	2.6	3	2			1		3	2	2.5	2.6	3	3

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 – L3, CO5 – L4 and CO6 – L5

Future Course Mapping:

Computational Linguistics, ANN, RNN, Deep Learning

Job Mapping:

Application Developers, System programmer.

Syllabus Template **FF No. : 654**

CS4275: Introduction to Machine Learning
(To be completed from NPTEL/SWAYAM)

Credits: 3

Course Prerequisites:
Statistical Mathematics, Artificial Intelligence

Course Relevance:

With the increased availability of data from varied sources there has been increasing attention paid to the various data driven disciplines such as analytics and machine learning. In this course we intend to introduce some of the basic concepts of machine learning from a mathematically well motivated perspective. We will cover the different learning paradigms and some of the more popular algorithms and architectures used in each of these paradigms..

All students have to complete this course from NPTEL-SWAYAM under the guidelines provided by dean academics and to be coordinated by department coordinator assigned for this course.

Syllabus Template **FF No. : 654**

CS4276: Deep Learning
(To be completed from NPTEL/SWAYAM)

Credits: 3

Course Prerequisites:

Statistical Mathematics, Artificial Intelligence, Machine Learning

Course Relevance:

The availability of huge volume of Image and Video data over the internet has made the problem of data analysis and interpretation a really challenging task. Deep Learning has proved itself to be a possible solution to such Computer Vision tasks. Not only in Computer Vision, Deep Learning techniques are also widely applied in Natural Language Processing tasks. In this course we will start with traditional Machine Learning approaches, e.g. Bayesian Classification, Multilayer Perceptron etc. and then move to modern Deep Learning architectures like Convolutional Neural Networks, Auto encoders etc. On completion of the course students will acquire the knowledge of applying Deep Learning techniques to solve various real life problems.

All students have to complete this course from NPTEL-SWAYAM under the guidelines provided by dean academics and to be coordinated by department coordinator assigned for this course.

Syllabus Template **FF No. : 654**

CS4225/CS4226 : Major Project

Credits:.13....... **Teaching Scheme Theory: …26... Hours/Week**

Course Prerequisites: Project Based Learning

Aim

This course addresses the issues associated with the successful management of a project. The course emphasizes project life cycle phases requirement engineering, system analysis and system design. A further aim is for students to heighten personal awareness of the importance of developing strategies for themselves and working with peers to create desired outcomes. The Project Work can lead to:

- Transform existing Ideas into conceptual models.
- Transform conceptual models into determinable models.
- Use determinable models to obtain system specifications.
- Select optimum specifications and create physical models.
- Apply the results from physical models to create real target systems.

Project Group and Topic Selection and Synopsis:

The project work needs to be undertaken by a group of maximum FOUR and minimum of THREE students. The Project work will be jointly performed by the project team members. The student needs to identify a technological problem in the area of Computer Engineering or Information Technology of their choice and address the problem by formulating a solution for the identified problem. The Project Group will prepare a synopsis of the project work which will be approved by the concerned faculty member. The project should not be a reengineering or reverse engineering project. In some cases, reverse engineering projects will be permissible based on the research component involved in it. The project work aims at solving a real world technical problem. Hence ample literature survey is required to be done by the students. Application-oriented projects will not be acceptable. Low-level custom User Interface development and its allied mapping with a particular technology will not be accepted.

Overview of the Course:

1. The Student Project Group is expected to make a survey of situation for identifying the requirements of selected Technological Problem. The Student Project Group will be monitored by Internal Guides and External Guides (if any).
2. The project requires the students to conceive, design, implement and operate a mechanism (the design problem). The mechanism may be entirely of the student’s own design, or it may incorporate off-the-shelf parts. If the mechanism incorporates off-the-shelf parts, the students

must perform appropriate analysis to show that the parts are suitable for their intended purpose in the mechanism.

3. The project must be based on a Fresh Idea or Implementation of a Theoretical Problem – meaning that there is not a known Solution to the design problem Or Create a Better Solution.
4. The project must have an experimental component. Students must conceive, design, implement and operate an appropriate experiment as part of the project. The experiment might be to collect data about some aspect of the design (i.e., to verify that the design will work as expected). Alternatively, the experiment could be to verify that the final mechanism performs as expected.
5. Upon receiving the approval, the Student Project Group will prepare a preliminary project report consisting , Feasibility Study Document, System Requirement Specification, System Analysis Document, Preliminary System Design Document. All the documents indicated will have a prescribed format.

6. Upon project completion, the Student Project Group will prepare a detailed Project Report consisting Semester I Preliminary Project document along with Detailed System Design Document, Implementation and Testing Document with conclusion and future scope of the Project Work. All the documents indicated will have a prescribed format. The Project Report ideally should consist of following documents : (Exceptions may be there based on the nature of the project, especially if some of the following documents are not applicable to a particular project as determined by the project guide, coordinator and head of department).

Sr.	Project Item
1	Project Cover Front Page
2	Project Completion Certificate [Institute]
3	Project Completion Letter [In case of Sponsored Projects]
4	Acknowledgments
5	Table of Contents
6	List of Figures
7	List of Tables
8	Project Synopsis [Problem Background, Existing System Details, Proposed Solution]
9	Feasibility Study Report
10	Project Plan
11	System Requirement Specification
12	System Analysis Document: UML Use Case Diagrams
13	System Analysis Document: UML Sequence Diagrams
14	System Analysis Document: UML State Diagrams

15	System Design Document with Module Specifications
16	System Implementation
17	System Testing and Experimental Findings
18	Conclusion
19	References

7. The Project Work will be assessed jointly by a panel of examiners consisting faculty and industry experts. The Project Groups will deliver the presentation and demonstration of the Project Work which will be assessed by the panel.
8. The Student Project Group needs to actively participate in the presentation. The panel of examiners will evaluate the candidate’s performance based on presentation skills, questions based on the Project Work and overall development effort taken by the candidates.

Note:
The student needs to design and develop solution for the identified technological problem in the area of Computer Engineering or Information Technology of their choice. The Project Implementation needs to be completed using best possible use of available technologies as applicable to deal with the complexity of the project. The Project Group will prepare a detailed report of the project work which will be approved by the concerned faculty member. The Project Report need to be submitted both in Hard form and Soft form in CD. The Soft Copy of the Project Report must accompany other project deliverables as well.

Assessment: MSE and ESE
1. Mid Semester Assessment – 50 Marks to be converted to 30 Marks.
2. End Semester Assessment – 100 Marks to be converted to 70 Marks.

Mid Semester Assessment

Sr. No.	Parameter	Marks
1	Problem Statement	10
2	Literature Review	10
3	Group formation and identification of individual responsibility	10
4	Objective of Project activity	10
5	Knowledge of domain, latest technology and modern tools used /to be used	10
	TOTAL	**50**

End Semester Assessment

Sr. No.	Parameter	Marks
1	Realization of project as per problem statement	10
2	Design, Testing / Experimentation, Analysis / Validation	30
3	Documentation and Report Writing	20
4	Quality of Work	15
5	Performance in Question & Answers Session	15
6	Regular interaction with guide	10
	TOTAL	**100**

Course Outcomes:

Upon completion of the course, graduates will be able to -

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

CO2: Prepare the requirement engineering, feasibility analysis documents

CO3: Form the teams and share responsibilities according to individual skill strengths

CO4: Create design documents to build software solutions

CO5: Develop software solutions based on standard engineering specifications

CO6: Perform the verification and validation up to the mark

CO PO Map

	Program Outcomes (PO)												PSO			
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**	**PSO4**
CO1	2	3		2					3			3				3
CO2	2	3	3	2	2				3	3	2	3	3		3	3
CO3	2	-	-	-	-				3		2	3	3			3
CO4	2	3	3	2	2	3	3.0	2.0	3	3	2	3	3	3	3	3
CO5	2	3	3	2	2				3	3	2	3	3	3	3	3
CO6	2	2	2	3	2				3	2	2	3	3	2	3	3
Average	**2.0**	**2.8**	**2.75**	**2.83**	**2.0**	**3.0**	**3.0**	**2.0**	**3.0**	**2.75**	**2.0**	**3.0**	**3.0**	**1.75**	**3.0**	**3.0**

CO attainment levels

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

Syllabus Template FF No. : 654

CS4232, CS4234, CS4202: Industry Internship, International Internship, Research Internship

Credits:.20...... Teaching Scheme Theory: …32... Hours/Week

Guidelines:

Internships are educational and career development opportunities, providing practical experience in a field or discipline. Internships are far more important as the employers are looking for employees who are properly skilled and having awareness about industry environment, practices and culture. Internship is structured, short-term, supervised training often focused around particular tasks or projects with defined time scales. Core objective is to expose technical students to the industrial environment, which cannot be simulated/experienced in the classroom and hence creating competent professionals in the industry and to understand the social, economic and administrative considerations that influence the working environment of industrial organizations. Engineering internships are intended to provide students with an opportunity to apply theoretical knowledge from academics to the realities of the field work/training.

The following guidelines are proposed to give academic credits for the internship undergone as a part of the B.Tech. Engineering curriculum.

Duration:

Industry Internship will be started at the beginning of the semester 7 or semester 8 or yearlong for the duration 6 months or 12 months.

Identification of Internship work:

Student may choose to undergo Internship at Industry/Govt./NGO/MSME/ Innovation/IPR/Entrepreneurship. Contacting various companies for Internship and Internship work identification process should be initiated at the end of 6th semester in coordination with training and placement cell/ industry institute cell/ internship cell. This will help students to start their internship work on time. Students can take internship work in the form of online/onsite work from any of the following but not limited to:

- Working for consultancy/ research project
- Contribution in Incubation/ Innovation/ Entrepreneurship Cell/ Institutional Innovation Council/ start-ups cells of institute
- Industry / Government Organization Internship,
- Internship through Internshala

- In-house product development, intercollegiate, inter department research internship under research lab/group, micro/small/medium enterprise/online internship
- Research internship under professors, IISC, IIT's, Research organizations

Internship Documents Submission:

Students must submit internship offer letter, internship completion letter, FF 1029 (Students feedback form), FF 1030 (Industry feedback about interns).

Students must present their internship progress time to time to faculty mentors. Faculty mentors and industry mentors both can evaluate the progress of the intern combiningly.

Internship Work Evaluation:

In-semester and end semester internship evaluation and assessment will be done by internal (Faculty mentor) and external examiners - a supervisor from industry.

After completion of Internship, the student should prepare a comprehensive report to indicate what he has observed and learnt in the internship/training period. The student may contact Industrial Supervisor/ Faculty Mentor/TPO for assigning special topics and problems and should prepare the final report.

If the student remain absent without prior intimation to the department/institute/concern authority/T & P Cell, his entire training can be cancelled and he will fail.

Course Outcomes: Industry Internship

On the completion of course, students will able to-

1. Understand real-world applications, workplace environment and operating procedures
2. Adapt skill for learning and applying modern tools and technologies
3. Apply professional values and ethical standards
4. Perform as an individual and as a team member effectively to changing conditions
5. Encompass improved writing, verbal communication and documentation skills
6. Learn about career positions and occupations along with the qualities and training required to obtain those positions

CO-PO Map: Industry Internship

	Program Outcomes (PO)												PSO			
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2	3		2					3			3				2
CO2	2	3	3	2	3	2	3	2	3		2	3	3	3	3	3
CO3	2	3	3	2				2	3		2	3	3	3	3	3
CO4	2								3			3				
CO5	2		2		3			2	3	3	2	3			3	
CO6	2					2			3			3				
Average	2.0	3.0	2.66	2.0	3.0	2.0	3.0	2.0	3.0	3.0	2.0	3.0	3.0	3.0	3.0	2.66

CO attainment levels:

CO1 – 3, CO2 –5, CO3 –2, CO4 –2, CO5 – 3, CO6 –2

Course Outcomes: Research Internship

On the completion of course, students will able to-

1. Develop an ecosystem to promote entrepreneurship and research culture among the students.
2. Learn first-hand to apply techniques, resources, modern engineering tools for prediction modelling to complex engineering activities.
3. Apply reasoning informed by the contextual knowledge to access societal, health, safety, legal and cultural issues and consequent responsibilities.
4. Perform as an individual and as a team member.
5. Understand Engineering and Management Principles.
6. Exercise R & D aptitude focusing on the knowledge creation and dissemination through engineering artifacts creation, construction and presentation.

CO-PO Map: Research Internship

	Program Outcomes (PO)												PSO			
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	2											3				3
CO2	3	3	3	2	3	2	2	2				3	3	3	3	3
CO3	3	3	3	2		3						3	3		3	3
CO4									3			3				3
CO5	3	2	3	2								3	3		3	3
CO6	3	3	2	3			3	2.0		3.0	2.0	3			3	3
Average	2.8	2.75	2.75	233	3.0	2.5	2.5	2.0	3.0	2.75	2.0	3.0	3.0	3.0	3.0	3.0

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