CURRICULUM

FOR

B. TECH DEGREE PROGRAMME

IN

COMPUTER SCIENCE AND ENGINEERING

2020 SCHEME (Revised in 2022)

(AUTONOMOUS)

MAR BASELIOS COLLEGE OF ENGINEERING AND TECHNOLOGY
(Approved by AICTE, Autonomous Institution Affiliated to APJ Abdul Kalam Technological University)
MAR IVANIOS VIDYANAGAR, NALANCHIRA, THIRUVANANTHAPURAM – 695015, KERALA.
Phone: 0471 2545866
Fax: 0471 2545869
Web: www.mbcet.ac.in
email: hodcs@mbcet.ac.in

CURRICULUM AND DETAILED SYLLABI

FOR

B. TECH DEGREE PROGRAMME

IN

COMPUTER SCIENCE AND ENGINEERING

2020 SCHEME (Revised in 2022)

(AUTONOMOUS)

MAR BASELIOS COLLEGE OF ENGINEERING AND TECHNOLOGY
(Approved by AICTE, Autonomous Institution Affiliated to APJ Abdul Kalam Technological University)
MAR IVANIOS VIDYANAGAR, NALANCHIRA, THIRUVANANTHAPURAM – 695015, KERALA.
Phone: 0471 2545866
Fax: 0471 2545869
Web: www.mbcet.ac.in
email: hodcs@mbcet.ac.in

MAR BASELIOS COLLEGE OF ENGINEERING AND TECHNOLOGY

Vision and Mission of the Institution

Vision:
To be an Institution moulding globally competent professionals as epitomes of Noble Values.

Mission:
To transform the Youth as technically competent, ethically sound and socially committed professionals, by providing a vibrant learning ambience for the welfare of humanity.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Vision and Mission of the Department

Vision:
To be a Centre of Excellence in Computer Science and Engineering providing quality education and research for the betterment of the society.

Mission:
To impart sound knowledge in theoretical and applied foundations of Computer Science and Engineering, and to train the students to solve real life issues to effectively define and shape life.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. Graduates will be successful professionals in Industries of core or interdisciplinary nature or entrepreneurs, demonstrating effective leadership and excellent team work.
2. Graduates will expand the horizon of knowledge through higher education or research, leading to self-directed professional development.
3. Graduates will demonstrate professional attitude and ethics while providing solutions in societal and environmental contexts.

PROGRAMME OUTCOMES (POs)

Engineering graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **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.
3. **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.
4. **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.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
6. **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.
7. **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.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **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.
11. **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.
12. **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.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

Engineering Graduates will have the ability to:

1. Apply Algorithmic Principles, Programming Skills and Software Engineering Principles to design, develop and evaluate Software Systems of varying complexities.
2. Apply knowledge of System Integration to design and implement computer-based systems.
3. Solve real world and socially relevant problems with the knowledge in recent and advanced Computing Technologies.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

For the students admitted from 2022-23

Scheduling of Courses

i) Knowledge Segments and Credits

Every course of B. Tech Programme is placed in one of the nine categories as listed in the following table.No semester shall have more than six lecture-based courses and two laboratory courses, and/or drawing/seminar/project courses in the curriculum.

Sl. No.	Category	Category Code	Total credits
1	Humanities and Social Sciences including Management Courses	HSC	5
2	Basic Science Courses	BSC	26
3	Engineering Science Courses	ESC	22
4	Programme Core Courses, Comprehensive Course Work and Viva Voce	PCC	79
5	Programme Elective Courses	PEC	15
6	Open Elective Courses	OEC	3
7	Project Work and Seminar	PWS	10
8	Mandatory Non-credit Courses (P/F) with Grade	MNC	---
9	Mandatory Student Activities (P/F)	MSA	2
Total Mandatory Credits			**162**
	Value Added Courses (Optional) – Honours/Minor	VAC	20

ii) Semester-wise Credit Distribution

Semester	**I**	**II**	**III**	**IV**	**V**	**VI**	**VII**	**VIII**	***Total Credits***
Credits for Courses	17	21	22	22	23	23	15	17	**160**
Activity Points (Min.)	40				60				**100**
Credits for Activities	2								**2**
Total Credits									**162**
Value Added Courses (Optional) – Honours / Minor									**20**
Total Credits									**182**

SEMESTER I						
Slot	**Category Code**	**Course Number**	**Courses**	**L-T-P**	**Hours**	**Credit**
A	BSC	MA0U10A	Linear Algebra and Calculus	3-1-0	4	4
B 1/2	BSC	PH0U10A	Engineering Physics-A	3-1-0	4	4
		CY0U10A	Engineering Chemistry-A	3-1-0	4	4
C 1/2	ESC	ES0U10A	Engineering Mechanics	2-1-0	3	3
		ES0U10B	Engineering Graphics	2-0-2	4	3
D 1/2	ESC	ES0U10C	Basics of Civil and Mechanical Engineering	4-0-0	4	4
		ES0U10D	Basics of Electrical and Electronics Engineering	4-0-0	4	4
E	HSC	HS0U10A	Life Skills	2-0-2	4	---
S 1/2	BSC	PH0U18A	Engineering Physics Lab	0-0-2	2	1
		CY0U18A	Engineering Chemistry Lab	0-0-2	2	1
T 1/2	ESC	ES0U18A	Civil and Mechanical Workshop	0-0-2	2	1
		ES0U18B	Electrical and Electronics Workshop	0-0-2	2	1
TOTAL					**23/24**	**17**

SEMESTER II						
Slot	**Category Code**	**Course Number**	**Courses**	**L-T-P**	**Hours**	**Credit**
A	BSC	MA0U10B	Vector Calculus, Differential Equations and Transforms	3-1-0	4	4
B 1/2	BSC	PH0U10A	Engineering Physics-A	3-1-0	4	4
		CY0U10A	Engineering Chemistry-A	3-1-0	4	4
C 1/2	ESC	ES0U10A	Engineering Mechanics	2-1-0	3	3
		ES0U10B	Engineering Graphics	2-0-2	4	3
D 1/2	ESC	ES0U10C	Basics of Civil and Mechanical Engineering	4-0-0	4	4
		ES0U10D	Basics of Electrical and Electronics Engineering	4-0-0	4	4
E	HSC	HS0U10B	Professional Communication	2-0-2	4	---
F	ESC	ES0U10F	Introduction to Computer Programming	2-1-2	5	4
S 1/2	BSC	PH0U18A	Engineering Physics Lab	0-0-2	2	1
		CY0U18A	Engineering Chemistry Lab	0-0-2	2	1
T 1/2	ESC	ES0U18A	Civil and Mechanical Workshop	0-0-2	2	1
		ES0U18B	Electrical and Electronics Workshop	0-0-2	2	1
TOTAL					**28/29**	**21**

SEMESTER III						
Slot	**Category Code**	**Course Number**	**Courses**	**L-T-P**	**Hours**	**Credit**
A	BSC	MA0U20G	Discrete Mathematical Structures	3-1-0	4	4
B	PCC	CS1U20A	Data Structures	3-1-0	4	4
C	PCC	CS1U20B	Logic System Design	3-1-0	4	4
D	PCC	CS1U20C	Object Oriented Programming using Java	3-1-0	4	4
E 1/2	ESC	ES0U20A	Design and Engineering	2-0-0	2	2
	HSC	HS0U20A	Professional Ethics	2-0-0	2	2
F	MNC	NC0U20A	Sustainable Engineering	2-0-0	2	---
S	PCC	CS1U28A	Data Structures Lab	0-0-3	3	2
T	PCC	CS1U28B	Object Oriented Programming Lab (in Java)	0-0-3	3	2
R/M	VAC		Remedial/Minor Course	3-1-0	4	4
TOTAL					**26/30**	**22/26**

SEMESTER IV						
Slot	**Category Code**	**Course Number**	**Courses**	**L-T-P**	**Hours**	**Credit**
A	BSC	MA0U20D	Probability Statistics and Numerical Methods	3-1-0	4	4
B	PCC	CS1U20D	Computer Organization and Architecture	3-1-0	4	4
C	PCC	CS1U20E	Database Management Systems	3-1-0	4	4
D	PCC	CS1U20G	Formal Languages and Automata Theory	3-1-0	4	4
E 1/2	ESC	ES0U20A	Design and Engineering	2-0-0	2	2
	HSC	HS0U20A	Professional Ethics	2-0-0	2	2
F	MNC	NC0U20B	Constitution of India	2-0-0	2	---
S	PCC	CS1U28C	Digital Lab	0-0-3	3	2
T	PCC	CS1U28E	Database Management Systems Lab	0-0-3	3	2
R/M /H	VAC		Remedial/Minor/Honours Course	3-1-0	4	4
TOTAL					**26/30**	**22/26**

SEMESTER V						
Slot	**Category Code**	**Course Number**	**Courses**	**L-T-P**	**Hours**	**Credit**
A	PCC	CS1U30J	Algorithm Analysis and Design	3-1-0	4	4
B	PCC	CS1U30C	System Software	3-1-0	4	4
C	PCC	CS1U30K	Operating Systems	3-1-0	4	4
D	PCC	CS1U30D	Microprocessors and Microcontrollers	3-1-0	4	4
E	PCC	CS1U30E	Management of Software Systems	3-0-0	3	3
F	MNC	NC0U30A	Disaster Management	2-0-0	2	---
S	PCC	CS1U38A	System Software and Microprocessors Lab	0-0-4	4	2
T	PCC	CS1U38D	Operating Systems Lab	0-0-4	4	2
R/M/H	VAC		Remedial/Minor/Honours Course	3-1-0	4	4
TOTAL					**29/33**	**23/27**

SEMESTER VI						
Slot	**Category Code**	**Course Number**	**Courses**	**L-T-P**	**Hours**	**Credit**
A	PCC	CS1U30F	Compiler Design	3-1-0	4	4
B	PCC	CS1U30G	Computer Graphics and Image Processing	3-1-0	4	4
C	PCC	CS1U30B	Computer Networks	3-1-0	4	4
D	PEC	CS1UXXX	Programme Elective I	2-1-0	3	3
E	HSC	HS0U30A	Industrial Economics and Foreign Trade	3-0-0	3	3
F	PCC	CS1U30I	Comprehensive Course Work	1-0-0	1	1
S	PCC	CS1U38C	Networking Lab	0-0-3	3	2
T	PWS	CS1U39A	Mini Project	0-0-3	3	2
R/M/H	VAC		Remedial/Minor/Honours Course	3-1-0	4	4
TOTAL					**25/29**	**23/27**

PROGRAMME ELECTIVE I

Slot	Category Code	Course Number	Courses	L-T-P	Hours	Credit
D	PEC	CS1U31A	Foundations of Machine Learning	2-1-0	3	3
		CS1U31B	Data Analytics	2-1-0	3	3
		CS1U31C	Foundations of Security in Computing	2-1-0	3	3
		CS1U31D	Automated Verification	2-1-0	3	3
		CS1U31E	Programming in Python	2-1-0	3	3
		CS1U31F	Advanced Data Communication	2-1-0	3	3

SEMESTER VII						
Slot	**Category Code**	**Course Number**	**Courses**	**L-T-P**	**Hours**	**Credit**
A	PCC	CS1U40A	Artificial Intelligence	2-1-0	3	3
B	PEC	CS1UXXX	Programme Elective II	2-1-0	3	3
C	OEC	CS0UXXX	Open Elective	2-1-0	3	3
D	MNC	NC0U40A	Industrial Safety Engineering	2-1-0	3	---
E	PCC	CS1U48A	Compiler Design Lab	0-0-3	3	2
T	PWS	CS1U49A	Seminar	0-0-3	3	2
U	PWS	CS1U49B	Project Phase I	0-0-6	6	2
R/M/H	VAC		Remedial/Minor/Honours Course	0-1-6/ 3-1-0	7/4	4
TOTAL					**24 (31/28)**	**15/19**

PROGRAMME ELECTIVE II

Slot	Category Code	Course Number	Course	L-T-P	Hours	Credit
B	PEC	CS1U41A	Machine Learning	2-1-0	3	3
		CS1U41B	Cloud Computing	2-1-0	3	3
		CS1U41C	Security in Computing	2-1-0	3	3
		CS1U41D	Model Based Software Development	2-1-0	3	3
		CS1U41E	Web Programming	2-1-0	3	3
		CS1U41F	Natural Language Processing	2-1-0	3	3

OPEN ELECTIVE I

Slot	Category Code	Course Number	Courses	L-T-P	Hours	Credit
C	OEC	CS0U41A	Introduction to Mobile Computing	2-1-0	3	3
		CS0U41B	Introduction to Deep Learning	2-1-0	3	3
		CS0U41C	Computer Graphics	2-1-0	3	3
		CS0U41D	Python for Engineers	2-1-0	3	3
		CS0U41E	Object Oriented Concepts	2-1-0	3	3

SEMESTER VIII						
Slot	**Category Code**	**Course Number**	**Courses**	**L-T-P**	**Hours**	**Credit**
A	PCC	CS1U40B	Distributed Computing	2-1-0	3	3
B	PEC	CS1UXXX	Programme Elective III	2-1-0	3	3
C	PEC	CS1UXXX	Programme Elective IV	2-1-0	3	3
D	PEC	CS1UXXX	Programme Elective V	2-1-0	3	3
T	PCC	CS1U40C	Comprehensive Viva Voce	1-0-0	1	1
U	PWS	CS1U49C	Project Phase II	0-0-12	12	4
R/M/H	VAC		Remedial/Minor/Honours Course	0-1-6	7	4
TOTAL					**25/32**	**17/21**

PROGRAMME ELECTIVE III

Slot	Category Code	Course Number	Courses	L-T-P	Hours	Credit
B	PEC	CS1U42A	Deep Learning	2-1-0	3	3
		CS1U42B	Programming Paradigms	2-1-0	3	3
		CS1U42C	Cryptography	2-1-0	3	3
		CS1U42D	Soft Computing	2-1-0	3	3
		CS1U42E	Fuzzy Set Theory and Application	2-1-0	3	3
		CS1U42F	Embedded Systems	2-1-0	3	3
		CS1U42G	Computer Vision	2-1-0	3	3

PROGRAMME ELECTIVE IV

Slot	Category Code	Course Number	Courses	L-T-P	Hours	Credit
C	PEC	CS1U43A	Formal Methods and Tools in Software Engineering	2-1-0	3	3
		CS1U43B	Client Server Architecture	2-1-0	3	3
		CS1U43C	Parallel Computing	2-1-0	3	3
		CS1U43D	Data Compression Techniques	2-1-0	3	3
		CS1U43F	Data Mining	2-1-0	3	3
		CS1U43G	Mobile Computing	2-1-0	3	3

PROGRAMME ELECTIVE V

Slot	Category Code	Course Number	Courses	L-T-P	Hours	Credit
D	PEC	CS1U44A	High Performance Computing	2-1-0	3	3
		CS1U44B	Blockchain Technologies	2-1-0	3	3
		CS1U44C	Image Processing Technique	2-1-0	3	3
		CS1U44D	Internet of Things	2-1-0	3	3
		CS1U44E	Software Testing	2-1-0	3	3
		CS1U44F	Bioinformatics	2-1-0	3	3
		CS1U44G	Computational Linguistics	2-1-0	3	3

MINOR

Semester	BUCKET I Specialization: SOFTWARE ENGINEERING				BUCKET II Specialization: MACHINE LEARNING				BUCKET III Specialization: NETWORKING			
	Course Number	**Course**	**L-T-P**	**Credit**	**Course Number**	**Course**	**L-T-P**	**Credit**	**Course Number**	**Course**	**L-T-P**	**Credit**
S3	CS0M 20A	Object Oriented Programming	3-1-0	4	CS0M 20B	Python for Machine Learning	3-1-0	4	CS0M 20C	Data Communication	3-1-0	4
S4	CS0M 20D	Programming Methodologies	3-1-0	4	CS0M 20E	Mathematics for Machine Learning	3-1-0	4	CS0M 20F	Introduction to Computer Networks	3-1-0	4
S5	CS0M 30A	Concepts in Software Engineering	3-1-0	4	CS0M 30B	Concepts in Machine Learning	3-1-0	4	CS0M 30C	Client Server Systems	3-1-0	4
S6	CS0M 30D	Introduction to Software Testing	3-1-0	4	CS0M 30E	Concepts in Deep Learning	3-1-0	4	CS0M 30F	Wireless Networks and IoT Applications	3-1-0	4
S7	CS0M 49A	Mini Project	0-1-6	4	CS0M 49A	Mini Project	0-1-6	4	CS0M 49A	Mini Project	0-1-6	4
S8	CS0M 49B	Mini Project	0-1-6	4	CS0M 49B	Mini Project	0-1-6	4	CS0M 49B	Mini Project	0-1-6	4

HONOURS

Semester	BUCKET I Specialization: SECURITY IN COMPUTING				BUCKET II Specialization: MACHINE LEARNING				BUCKET III Specialization: FORMAL METHODS			
	Course Number	**Course**	**L-T-P**	**Credit**	**Course Number**	**Course**	**L-T-P**	**Credit**	**Course Number**	**Course**	**L-T-P**	**Credit**
S4	CS1H 20A	Number Theory	3-1-0	4	CS1H 20B	Computational Fundamentals of Machine Learning	3-1-0	4	CS1H 20C	Principles of Program Analysis and Verification	3-1-0	4
S5	CS1H 30A	Cryptographic Algorithms	3-1-0	4	CS1H 30B	Neural Networks and Deep Learning	3-1-0	4	CS1H 30C	Principles of Model Checking	3-1-0	4
S6	CS1H 30D	Network Security	3-1-0	4	CS1H 30E	Advanced Topics in Machine Learning	3-1-0	4	CS1H 30F	Theory of Computability and Complexity	3-1-0	4
S7	CS1H 40A	Cyber Forensics	3-1-0	4	CS1H 40B	Reinforcement Learning	3-1-0	4	CS1H 40C	Logic for Computer Science	3-1-0	4
S8	CS1H 49A	Mini Project	0-1-6	4	CS1H 49A	Mini Project	0-1-6	4	CS1H 49A	Mini Project	0-1-6	4

SEMESTER VII

CS1U40A	ARTIFICIAL INTELLIGENCE	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	2	1	0	3	2020

COURSE OVERVIEW

The course aims to introduce the fundamental principles of intelligent systems to students. This involves ideas about the characteristics of intelligent systems, knowledge representation schemes, logic and inference mechanisms. The course helps the learner to understand the design of self-learning systems along with some of their typical applications in the emerging scenario where the business world is being transformed by the progress made in machine learning.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Explain the fundamental concepts of intelligent systems and their architecture.	Understand
CO2	Make use of uninformed and informed search techniques for problem solving in intelligent systems.	Apply
CO3	Represent AI domain knowledge using logic systems and use inference techniques for reasoning in intelligent systems.	Apply
CO4	Solve Constraint Satisfaction Problems using search techniques.	Apply
CO5	Identify different types of learning techniques used in intelligent systems	Apply

SYLLABUS

Introduction: What is Artificial Intelligence(AI), The Foundations of AI, History of AI, Applications of AI. Intelligent Agents – Agents and Environments, Good behavior: The concept of rationality, nature of Environments, Structure of Agents.

Problem Solving: Solving Problems by searching-Problem solving Agents, Example problems, Searching for solutions, Uninformed search strategies, Informed search strategies, Heuristic functions.

Knowledge Representation and Reasoning

Logical Agents: Knowledge based agents, Logic, Propositional Logic, Propositional Theorem proving, Agents based on Propositional Logic. First Order Predicate Logic – Syntax and Semantics of First Order Logic, Using First Order Logic, Knowledge representation in First Order Logic. Inference in First Order Logic – Propositional Vs First Order inference, Unification and Lifting, Forward chaining, Backward chaining, Resolution.

Search in Complex environments: Adversarial search - Games, Optimal decisions in games, The Minimax algorithm, Alpha-Beta pruning. Constraint Satisfaction Problems – Defining CSP, Constraint Propagation-inference in CSPs, Backtracking search for CSPs, Structure of CSP problems.

Machine Learning: Learning from Examples – Forms of Learning, Supervised Learning, Learning Decision Trees, Evaluating and choosing the best hypothesis, Regression and classification with Linear models.

TEXT BOOK

1. Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, 3rd Edition. Prentice Hall.

REFERENCES

1. Nilsson N.J., Artificial Intelligence - A New Synthesis, Harcourt Asia Pvt. Ltd.

COURSE PLAN

Module	Contents	No. of hours
1	Introduction, What is Artificial Intelligence(AI)?, The foundations of AI, The history of AI, Applications of AI, Intelligent Agents – Agents and Environments, Good behavior: The concept of rationality, The nature of Environments, The structure of Agents	7
2	Solving Problems by searching-Problem solving Agents, Illustration of the problem solving process by agents, Searching for solutions, Uninformed search strategies: BFS, Uniform-cost search, DFS, Depth-limited search, Iterative deepening depth-first search, Informed search strategies: A* Search, Heuristic functions.	11
3	Logical Agents – Knowledge based agents and logic, Propositional Logic, Propositional Theorem proving, Agents based on Propositional Logic, First Order Predicate Logic – Syntax and Semantics of First Order Logic, Using First Order Logic, Knowledge representation in First Order Logic, Inference in First Order Logic – Propositional Vs First Order inference, Unification and Lifting, forward chaining, Backward chaining, Resolution	9
4	Adversarial search – Games, Optimal decisions in games, The Minimax algorithm, Alpha-Beta pruning. Constraint Satisfaction Problems – Defining CSP, Constraint Propagation- inference in CSPs, Backtracking search for CSPs, The structure of problems.	9
5	Learning from Examples – Forms of Learning, Supervised Learning, Learning Decision Trees, Generalization and overfitting, Evaluating and choosing the best hypothesis, Regression and classification with Linear models.	9
	Total hours	**45**

NCOU40A	INDUSTRIAL SAFETY ENGINEERING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	2	1	0	-	2020

COURSE OVERVIEW

The course is intended to give knowledge of various safety management principles, various safety systems, various machine guarding devices, hazard identification techniques, energy sources, systems & applications and the need in the present context. Learners will be able to compare different hazard identification tools and choose the most appropriate based on the nature of industry. It aims to equip students in working with projects and to take up research work in connected areas.

COURSE OUTCOME

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Describe the theories of accident causation and preventive measures of industrial accidents.	Understand
CO2	Explain about personal protective equipment, its selection, safety performance & indicators and importance of housekeeping.	Understand
CO3	Explain different issues in construction industries.	Understand
CO4	Describe various hazards associated with different machines and mechanical material handling.	Understand
CO5	Utilize different hazard identification tools in different industries with the knowledge of different types of chemical hazards.	Understand

SYLLABUS

Safety introduction: Need for safety. Safety and productivity. Definitions: Accident, Injury, Unsafe act, Unsafe Condition, Dangerous Occurrence, Reportable accidents. Theories of accident causation. Safety organization- objectives, types, functions, Role of management, supervisors, workmen, unions, government and voluntary agencies in safety. Safety policy. Safety Officer-responsibilities, authority. Safety committee-need, types, advantages.

Personal protection in work environment: Personal protection in the work environment, Types of PPEs, Personal protective equipment- respiratory and non-respiratory equipment. Standards related to PPEs. Monitoring Safety Performance: Frequency rate, severity rate, incidence rate, activity rate. Housekeeping: Responsibility of management and employees. Advantages of good housekeeping. 5 s of housekeeping. Work permit system- objectives, hot work and cold work permits. Typical industrial models and methodology. Entry into confined spaces.

Safety issues in construction: Introduction to construction industry and safety issues in construction Safety in various construction operations – Excavation and filling – Under-water works – Under-pinning & Shoring – Ladders & Scaffolds – Tunneling – Blasting – Demolition – Confined space – Temporary Structures. Familiarization with relevant Indian Standards and the National Building Code provisions on construction safety. Relevance of ergonomics in construction safety. Ergonomics Hazards - Musculoskeletal Disorders and Cumulative Trauma Disorders.

Safety hazards in machines: Machinery safeguard-Point-of-Operation, Principle of machine guarding -types of guards and devices. Safety in turning, and grinding. Welding and Cutting-Safety Precautions of Gas welding and Arc Welding. Material Handling-Classification-safety consideration- manual and mechanical handling. Handling assessments and techniques- lifting, carrying, pulling, pushing, palletizing and stocking. Material Handling equipment-operation & maintenance. Maintenance of common elements-wire rope, chains slings, hooks, clamps. Hearing Conservation Program in Production industries.

Hazard identification and analysis: Hazard and risk, Types of hazards –Classification of Fire, Types of Fire extinguishers, fire explosion and toxic gas release, Structure of hazard identification and risk assessment. Identification of hazards: Inventory analysis, Fire and explosion hazard rating of process plants-The Dow Fire and Explosion Hazard Index, Preliminary hazard analysis, Hazard and Operability study (HAZOP)) – methodology, criticality analysis, corrective action and follow-up. Control of Chemical Hazards, Hazardous properties of chemicals, Material Safety Data Sheets (MSDS).

TEXT BOOKS:

1. R.K Jain (2000) Industrial Safety, Health and Environment management systems, Khanna Publications.
2. Paul S V (2000), Safety management System and Documentation training Programme handbook, CBS Publication.
3. Krishnan, N.V. (1997). *Safety management in Industry*. Jaico Publishing House, New Delhi.
4. John V. Grimaldi and Rollin H.Simonds. (1989) *Safety management*. All India Traveller Book Seller, Delhi.
5. Ronald P. Blake. (1973). *Industrial safety.* Prentice Hall, New Delhi.
6. Alan Waring. (1996). *Safety management system*. Chapman & Hall, England.
7. Vaid, K.N., (1988). Construction safety management. National Institute of Construction Management and Research, Mumbai.
8. AIChE/CCPS. (1992). *Guidelines for Hazard Evaluation Proce*dures. (second edition). Centre for Chemical Process Safety, American Institute of Chemical Engineers, New York.

REFERENCES

1. Ronald P. Blake. (1973). *Industrial safety*. Prentice Hall, New Delhi.
2. Alan Waring. (1996). *Safety management system*. Chapman & Hall, England.
3. Vaid, K.N., (1988). Construction safety management. National Institute of Construction Management and Research, Mumbai.
4. AIChE/CCPS. (1992). *Guidelines for Hazard Evaluation Procedures*. (second edition). Centre for Chemical Process Safety, American Institute of Chemical Engineers, New York.

COURSE PLAN

Module	Contents	No. of hours
1	**Introduction to Industrial safety Engineering** Need for safety. Safety and productivity. Definitions: Accident, Injury, Unsafe act, Unsafe Condition, Dangerous Occurrence. Reportable accidents, Theories of accident causation. Safety organization. Role of management, supervisors, workmen, unions, government and voluntary agencies in safety. Safety officer-responsibilities, authority, Safety committee need, types, advantages.	9
2	**Personal protection in the work environment** Types of PPEs, respiratory and non-respiratory equipment, Standards related to PPEs, Monitoring Safety Performance: Frequency rate, severity rate, Monitoring Safety Performance: incidence rate, activity rate. Housekeeping: Responsibility of management and employees. Advantages of good housekeeping. 5 s of housekeeping, Work permit system-objectives, hot work and cold work permits, Typical industrial models and methodology. Entry into confined spaces, Types of PPEs, respiratory and non-respiratory equipment.	9
3	**Introduction to construction industry and safety** Excavation and filling – Under-water works – Under-pinning & Shoring, Ladders & Scaffolds – Tunneling, Blasting –Demolition – Confined space, Familiarization with relevant Indian Standards and the National Building Code provisions on construction safety. Relevance of ergonomics in construction safety. Ergonomics Hazards, Musculoskeletal Disorders and Cumulative Trauma Disorders.	9
4	**Machinery safeguard** Point-of-Operation, Principle of machine guarding, Types of guards and devices. Safety in Power Presses, primary & secondary operations – shearing-bending - rolling – drawing. Safety in turning, boring, milling, planning and grinding. Welding and Cutting-Safety Precautions of Gas welding and Arc Welding, Cutting and Finishing. Material Handling-Classification-safety consideration- manual and mechanical handling. Handling assessments and techniques- lifting, carrying, pulling, pushing, palletizing and stocking. Material Handling equipment-operation & maintenance. Maintenance of common elements-wire rope, chains slings, hooks, clamps	9
5	**Hazard identification** Hazard and risk, Types of hazards – Classification of Fire, Types of Fire extinguishers fire, explosion and toxic gas release. Inventory analysis, Fire and explosion hazard rating of process plants, The Dow Fire and Explosion Hazard Index. Preliminary hazard analysis, Hazard and Operability study (HAZOP). Chemical hazard- Classifications, Control of Chemical Hazards. Hazardous properties of chemicals. Material Safety Data Sheets (MSDS).	9
	Total hours	**45**

CS1U48A	COMPILER DESIGN LAB	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	0	0	3	2	2020

COURSE OVERVIEW:

This course aims to offer students hands-on experience on compiler design concepts. Students will be able to familiarize with tools such as LEX and YACC and automate different phases of a compiler. This course helps the learners to enhance the capability to design and implement a compiler.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO 1	Implement lexical analyzer using the tool LEX.	Apply
CO 2	Implement Syntax analyzer using the tool YACC.	Apply
CO 3	Design NFA and DFA for a problem and write programs to perform operations on it.	Apply
CO 4	Design and Implement Top-Down parsers.	Apply
CO 5	Design and Implement Bottom-Up parsers.	Apply
CO 6	Implement intermediate code for expressions.	Apply

SYLLABUS

1. Implementation of lexical analyzer using the tool LEX.

2. Implementation of Syntax analyzer using the tool YACC.

3. Application problems using NFA and DFA.

4. Implement Top-Down Parser.

5. Implement Bottom-up parser.

6. Simulation of code optimization Techniques.

7. Implement Intermediate code generation for simple expressions.

8. Implement the back end of the compiler.

PRACTICE QUESTIONS

List of Exercises/Experiments:

1. Design and implement a lexical analyzer using C language to recognize all valid tokens in the input program. The lexical analyzer should ignore redundant spaces, tabs and newlines. It should also ignore comments.
2. Implement a Lexical Analyzer for a given program using Lex Tool.

3. Write a lex program to display the number of lines, words and characters in an input text.

4. Write a LEX Program to convert the substring *abc* to *ABC* from the given input string.

5. Write a lex program to find out total number of vowels and consonants from the given input sting.
6. Generate a YACC specification to recognize a valid arithmetic expression that uses operators +, –, *, / and parenthesis.
7. Generate a YACC specification to recognize a followed by any number of letters or digits.
8. Implementation of Calculator using LEX and YACC

9. Convert the BNF rules into YACC form and write code to generate abstract syntax tree.
10. Write a program to find ε – closure of all states of any given NFA with ε transition.

11. Write a program to convert NFA with ε transition to NFA without ε transition.

12. Write a program to convert NFA to DFA.

13. Write a program to minimize any given DFA.

14. Write a program to find First and Follow of any given grammar.

15. Design and implement a recursive descent parser for a given grammar.

16. Construct a Shift Reduce Parser for a given language.

17. Write a program to perform constant propagation.

18. Implement Intermediate code generation for simple expressions.

19. Implement the back end of the compiler which takes the three address code and produces the 8086 assembly language instructions that can be assembled and run using an 8086 assembler. The target assembly instructions can be simple move, add, sub, jump etc

TEXT BOOK

1. Aho A.V., Ravi Sethi and D. Ullman. Compilers – Principles Techniques and Tools,Addison Wesley, 2006.

REFERENCES

1. D.M.Dhamdhere, System Programming and Operating Systems, Tata McGraw Hill & Company, 1996.

2. Kenneth C. Louden, Compiler Construction – Principles and Practice, Cengage Learning Indian Edition, 2006.

3. Tremblay and Sorenson, The Theory and Practice of Compiler Writing, Tata McGraw Hill & Company,1984.

CS1U49A	SEMINAR	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PWS	0	0	3	2	2020

COURSE OVERVIEW:

The course 'Seminar' is intended to enable a B.Tech graduate to read, understand, present and prepare report about an academic document. The learner shall search in the literature including peer reviewed journals, conference, books, project reports etc., and identify an appropriate paper/thesis/report in her/his area of interest, in consultation with her/his seminar guide. This course can help the learner to experience how a presentation can be made about a selected academic document and also empower her/him to prepare a technical report.

COURSE OUTCOMES

After successful completion of the course, the students will be able to:

Course Outcomes	Description	Level
CO1	Identify academic documents from the literature which are related to her/his areas of interest.	Apply
CO2	Read and apprehend an academic document from the literature which is related to her/ his areas of interest	Analyze
CO3	Prepare a presentation about an academic document	Create
CO4	Give a presentation about an academic document	Apply
CO5	Prepare a technical report.	Create

General Guidelines

- The Department shall form an Internal Evaluation Committee (IEC) for the seminar with academic coordinator for that program as the Chairperson/Chairman and seminar coordinator & seminar guide as members. During the seminar presentation of a student, all members of IEC shall be present.

- Formation of IEC and guide allotment shall be completed within a week after the University examination (or last working day) of the previous semester.

- Guide shall provide required input to their students regarding the selection of topic/ paper.

- Choosing a seminar topic: The topic for a UG seminar should be current and broad based rather than a very specific research work. It's advisable to choose a topic for the Seminar to be closely linked to the final year project area. Every member of the project team could choose or be assigned Seminar topics that covers various aspects linked to the Project area.

- A topic/paper relevant to the discipline shall be selected by the student during the semester break.

- Topic/Paper shall be finalized in the first week of the semester and shall be submitted to the IEC.
- The IEC shall approve the selected topic/paper by the second week of the semester.
- Accurate references from genuine peer reviewed published material to be given in the report and to be verified.

EVALUATION PATTERN

Total marks: 100, only CIE, minimum required to pass 50

Seminar Guide: 20 marks (Background Knowledge – 10 (The guide shall give deserving marks for a candidate based on the candidate's background knowledge about the topic selected), Relevance of the paper/topic selected – 10).

Seminar Coordinator: 20 marks (Seminar Diary – 10 (Each student shall maintain a seminar diary and the guide shall monitor the progress of the seminar work on a weekly basis and shall approve the entries in the seminar diary during the weekly meeting with the student), Attendance – 10).

Presentation: 40 marks to be awarded by the IEC (Clarity of presentation – 10, Interactions – 10 (to be based on the candidate's ability to answer questions during the interactive session of her/his presentation), Overall participation – 10 (to be given based on her/his involvement during interactive sessions of presentations by other students), Quality of the slides – 10).

Report: 20 marks to be awarded by the IEC (check for technical content, overall quality, templates followed, adequacy of references etc.).

CS1U49B	PROJECT PHASE I	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PWS	0	0	6	2	2020

PREAMBLE

The course 'Project Work' is mainly intended to evoke the innovation and invention skills in a student. The course will provide an opportunity to synthesize and apply the knowledge and analytical skills learned, to be developed as a prototype or simulation. The project extends to 2 semesters and will be evaluated in the 7th and 8th semester separately, based on the achieved objectives. One third of the project credits shall be completed in 7th semester and two third in 8th semester. It is recommended that the projects may be finalized in the thrust areas of the respective engineering stream or as interdisciplinary projects. Importance should be given to address societal problems and developing indigenous technologies

COURSE OVERVIEW

The course 'Project Work' is mainly intended to evoke the innovation and invention skills in a student. The course will provide an opportunity to synthesize and apply the knowledge and analytical skills learned, to be developed as a prototype or simulation. The project extends to 2 semesters and will be evaluated in the 7th and 8th semester separately, based on the achieved objectives. One third of the project credits shall be completed in 7th semester and two third in 8th semester. It is recommended that the projects may be finalized in the thrust areas of the respective engineering stream or as interdisciplinary projects. Importance should be given to address societal problems and developing indigenous technologies.

COURSE OUTCOMES

After successful completion of the course, the students will be able to:

Course Outcomes	Description	Level
CO1	Model and solve real world problems by applying knowledge across domains	Apply
CO2	Develop products, processes or technologies for sustainable and socially relevant applications	Apply
CO3	Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks .	Apply
CO4	Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms .	Apply
CO5	Identify technology/research gaps and propose innovative/creative solutions.	Analyze
CO6	Organize and communicate technical and scientific findings effectively in written and oral forms .	Apply

Phase 1 Target

- Literature study/survey of published literature on the assigned topic
- Formulation of objectives
- Formulation of hypothesis/ design/ methodology
- Formulation of work plan and task allocation.
- Block level design documentation

- Seeking project funds from various agencies
- Preliminary Analysis/Modeling/Simulation/Experiment/Design/Feasibility study
- Preparation of Phase 1 report

Evaluation Guidelines & Rubrics

Total: 100 marks (Minimum required to pass: 50 marks).

- Project progress evaluation by guide: 30 Marks.
- Interim evaluation by the Evaluation Committee: 20 Marks.
- Final Evaluation by the Evaluation Committee: 30 Marks.
  Project Phase - I Report (By Evaluation Committee): 20 Marks

(The evaluation committee comprises HoD or a senior faculty member, Project coordinator and project supervisor).

PROGRAM ELECTIVE II

CS1U41A	MACHINE LEARNING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

This course enables the learners to understand the advanced concepts and algorithms in machine learning. The course covers the standard and most popular supervised learning algorithms such as linear regression, logistic regression, decision trees, Bayesian learning and the Naive Bayes algorithm, basic clustering algorithms and classifier performance measures. This course helps the students to provide machine learning based solutions to real world problems.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Illustrate Machine Learning concepts and basic parameter estimation methods.	Apply
CO2	Demonstrate supervised learning concepts (regression, Linear classification).	Apply
CO3	Illustrate the concepts of Multilayer neural network and Support Vector Machine	Apply
CO4	Describe unsupervised learning concepts and dimensionality reduction techniques.	Apply
CO5	Solve real life problems using appropriate machine learning models and evaluate the performance measures .	Apply

SYLLABUS

Overview of machine learning: Machine learning paradigms-supervised, semi-supervised, unsupervised, reinforcement learning. Basics of parameter estimation - maximum likelihood estimation(MLE) and maximum a posteriori estimation(MAP). Introduction to Bayesian formulation.

Supervised Learning: Regression - Linear regression with one variable, Linear regression with multiple variables, solution using gradient descent algorithm and matrix method, basic idea of overfitting in regression. Linear Methods for Classification- Logistic regression, Naive Bayes, Decision tree algorithm ID3.

Neural Networks (NN) and Support Vector Machines (SVM): Perceptron, Neural Network - Multilayer feed forward network, Activation functions (Sigmoid, ReLU, Tanh), Backpropagation algorithm.SVM - Introduction, Maximum Margin Classification, Mathematics behind Maximum Margin Classification, Maximum Margin linear separators, soft margin SVM classifier, non-linear SVM, Kernels for learning non-linear functions, polynomial kernel, Radial Basis Function(RBF).

Unsupervised Learning: Clustering - Similarity measures, Hierarchical Agglomerative Clustering, K-means partitional clustering, Expectation maximization (EM) for soft clustering. Dimensionality reduction –Principal Component Analysis.

Classification Assessment: Classification Performance measures - Precision, Recall, Accuracy, F-Measure, Receiver Operating Characteristic Curve(ROC), Area Under Curve(AUC. Bootstrapping, Cross Validation, Ensemble methods, Bias-Variance decomposition. Case Study: Develop a classifier for face detection.

COURSE PLAN

Module	Contents	No. of Hours
1	Supervised, semi-supervised, unsupervised learning, reinforcement learning, Maximum likelihood estimation(MLE), Maximum likelihood estimation (MLE)- example, Maximum a posteriori estimation(MAP), Maximum a posteriori estimation(MAP)-example, Bayesian formulation, Bayesian formulation –example	9
2	Linear regression with one variable, Multiple variables, Solution using gradient descent algorithm and matrix method (No derivation required), Overfitting in regression, Lasso and Ridge regularization, Logistic regression, Naive Bayes, Decision trees, Decision trees- ID3 algorithm.	9
3	Perceptron, Perceptron Learning, Multilayer Feed forward Network, Activation Functions (Sigmoid, ReLU,Tanh), Back Propagation Algorithm, Illustrative Example for Back Propagation, Introduction, Maximum Margin Hyperplane, Mathematics behind Maximum Margin Classification, Formulation of maximum margin hyperplane and solution, Soft margin SVM, Solution of Soft margin SVM, Non-linear SVM , Kernels for learning non-linear functions, Examples -Linear, RBF, Polynomial.	9
4	Similarity measures- Minkowski distance measures(Manhattan, Euclidean), Cosine Similarity, Clustering - Hierarchical Clustering, K-means partitional clustering, Expectation maximization (EM) for soft clustering, Expectation maximization (EM) for soft clustering, Dimensionality reduction – Principal Component Analysis, Dimensionality reduction – Principal Component Analysis.	9
5	Performance measures - Precision, Recall, Accuracy, F-Measure, ROC, AUC, Boot strapping, Cross validation, Ensemble methods- bagging, boosting, Bias-Variance decomposition, Bias-Variance decomposition, Face detection	9
	Total hours	**45**

TEXT BOOKS

1. Ethem Alpaydin, Introduction to Machine Learning, 2nd edition, MIT Press 2010.
2. Mohammed J. Zaki and Wagner Meira, Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, First South Asia edition, 2016.
3. Jake VanderPlas, Python Data Science Handbook, O'Reilly Media, 2016
4. Tom Mitchell, Machine Learning, McGraw-Hill, 1997.

REFERENCES

1. Christopher Bishop. Neural Networks for Pattern Recognition,Oxford University Press, 1995.
2. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective, MIT Press 2012.
3. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements Of Statistical Learning, Second edition Springer 2007.
4. P. Langley, Elements of Machine Learning, Morgan Kaufmann, 1995.
5. Richert and Coelho, Building Machine Learning Systems with Python.

6. Davy Cielen, Arno DB Meysman and Mohamed Ali.Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Dreamtech Press 2016.

CS1U41B	CLOUD COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW

This course helps the learners to understand cloud computing concepts. This course includes basic understanding of virtualization, fundamentals of cloud security, cloud computing based programming techniques and different industry popular cloud computing platforms. This course enables the student to suggest cloud based solutions to real world problems.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Explain the various cloud computing models and services.:	Understand
CO2	Demonstrate the significance of implementing virtualization techniques.	Understand
CO3	Explain different cloud enabling technologies and compare private cloud platforms.	Understand
CO4	Apply appropriate cloud programming methods to solve big data problems. (Cognitive Knowledge Level.	Apply
CO5	Describe the need for security mechanisms in cloud .	Understand
CO6	Compare the different popular cloud computing platforms.	Understand

SYLLABUS

Fundamental Cloud Computing: Traditional computing- Limitations. Overview of Computing Paradigms-Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing. NIST reference Model-Basic terminology and concepts. Cloud characteristics, benefits and challenges, Roles and Boundaries. Cloud delivery (service) Models-Infrastructure-as-a-Service (IaaS), Platform-as-a-Service(PaaS), Software-as-a-Service (SaaS), XaaS (Anything-as-a-service)-Cloud deployment models- Public cloud, Community cloud, Private cloud, Hybrid cloud.

Virtualization : Introduction to Virtualization-Virtualizing physical computing resources, Virtual Machines (Machine virtualization), non-virtualized v/s virtualized machine environments. Types of VMs-process VM v/s system VM, Emulation, interpretation and binary translation. Hardware-level virtualization- Hypervisors/VMM. Types of Hypervisors. Full Virtualization, Para-Virtualization, Hardware-assisted virtualization, OS level virtualization. Basics of Network Virtualization, Storage Virtualization and Desktop Virtualization, Pros and cons of virtualization. Case Study- Xen: Para-virtualization, VMware: full virtualization.

Cloud-Enabling Technologies, Private cloud platforms and programming: Broadband networks and internet architecture- Internet Service Providers (ISPs), Data center technology, Web technology, Multitenant technology, Service technology. Resource provisioning techniques-static and dynamic provisioning.Open-source software platforms for private Cloud-OpenStack, CloudStack, Basics of Eucalyptus, Open Nebula, Nimbus.Cloud Programming- Parallel Computing and Programming Paradigms. Map Reduce – Hadoop Library from Apache, HDFS, Pig Latin High Level Languages, Apache Spark.

Fundamental Cloud Security: Basic terms and concepts in security- Threat agents, Cloud security threats/risks, Trust. Operating system Security-Virtual machine security- Security of virtualization- Security Risks Posed by Shared Images, Security Risks Posed by Management OS. Infrastructure Security-Network Level Security, Host Level Security, Application level security, Security of the Physical Systems. Identity & Access Management- Access Control.

Popular Cloud Platforms : Amazon Web Services(AWS):- AWS ecosystem- Computing services, Amazon machine images, Elastic Compute Cloud (EC2), Advanced compute services. Storage services- Simple Storage System (Amazon S3), Elastic Block Store (Amazon EBS), Database Services, Amazon CDN Services and Communication services. **Google Cloud Platform**: - IaaS Offerings: Compute Engine (GCE), Cloud Storage, PaaS Offerings: Google App Engine (GAE), Storage services, Application services, Compute services, Database Services, SaaS Offerings: Gmail, Docs, Google Drive.

Microsoft Azure: Azure Platform Architecture, Hyper-V, Azure Virtual Machine, Compute services, Storage services.

TEXT BOOKS

1. Thomas, E., Zaigham M., Ricardo P "Cloud Computing Concepts, Technology & Architecture.", (2013 Edition). Prentice Hall.
2. Buyya, R., Vecchiola, C., & Selvi, S. T. "Mastering cloud computing: foundations and applications programming", (2017 Edition), Morgan Kaufmann.
3. Bhowmik, S., "Cloud computing", (2017 Edition). Cambridge University Press.

REFERENCES

1. Marinescu, D. C., "Cloud computing: theory and practice.", (2017 Edition). Morgan Kaufmann.
2. Buyya, R., Broberg, J., & Goscinski, A. M., "Cloud computing: Principles and paradigms" (2011 Edition). John Wiley & Sons.

COURSE PLAN

Module	Contents	No. of Hours
1	**Fundamental Cloud Computing** Traditional computing: Limitations, Overview of Computing Paradigms: Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing, NIST reference Model, Basic terminology and concepts, Cloud characteristics and benefits, challenges. Roles and Boundaries, Cloud delivery (service) models: Infrastructure-as-a-Service (IaaS), Platform-asa-Service (PaaS), Software-as-a-Service (SaaS), XaaS (Anything-as-a-service), Cloud deployment models: Public cloud, Community cloud, Private cloud, Hybrid cloud	9

2	**Virtualization** Introduction to virtualization, Virtualizing physical computing resources Virtual Machines (Machine virtualization):- non-virtualized v/s virtualized machine environments, Types of VMs: process VM v/s system VM, Emulation, interpretation and binary translation, Hardware-level virtualization: Hypervisors/VMM, Types of Hypervisors, Full Virtualization, Para-Virtualization, Hardware-assisted virtualization, OS level virtualization, Basics of Network Virtualization, Storage Virtualization and Desktop Virtualization, Pros and cons of virtualization, Case Study: Xen: Para-virtualization, Case Study: VMware: full virtualization	9
3	**Cloud-Enabling Technologies, Private cloud platforms and programming** Broadband networks and internet architecture: Internet Service Providers (ISPs), Data center technology, Web technology, Multitenant technology, Service technology., Resource provisioning techniques: static and dynamic provisioning, Open-source software platforms for private cloud: OpenStack, CloudStack, Basics of Eucalyptus, Open-Nebula, Nimbus, Cloud Programming: Parallel Computing and Programming Paradigms, Map Reduce, Hadoop Library from Apache, HDFS, Pig Latin High Level Languages, Apache Spark	9
4	**Fundamental Cloud Security** Basic terms and concepts in security, Threat agents, Cloud security threats/risks, Trust, Operating system security, Virtual machine security, Security of virtualization, Security Risks posed by Shared Images, Security Risks posed by Management OS, Infrastructure security: - Network Level Security, Host Level Security, Application level security, Security of the Physical Systems, Identity & Access Management, Access Control	9
5	**Popular Cloud Platforms** Amazon Web Services(AWS):- AWS ecosystem, Computing services: Amazon machine images, Elastic Compute Cloud (EC2), Advanced computing services, Storage services: Simple Storage System (Amazon S3), Elastic Block Store (Amazon EBS), Database Services, Amazon CDN Services and Communication services, Google Cloud Platform:- IaaS Offerings: Compute Engine (GCE), Cloud Storage, PaaS Offerings: Google App Engine (GAE), Storage services, Application services, Compute services, Database Services, SaaS Offerings: Gmail, Docs, Google Drive, Microsoft Azure: Azure Platform Architecture, Hyper-V, Azure Virtual Machine, Azure Compute services, Storage services	9
	Total hours	**45**

CS1U41C	SECURITY IN COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW

This course helps the learners to explore various algorithms to offer confidentiality, integrity, authentication &non-repudiation services and different attacks on system security with their countermeasures. It covers classical encryption techniques, symmetric and public key crypto-system, key distribution techniques, authentication functions, intruders, malicious software, and DDoS attacks. The concepts covered in this course enable the learners in effective use of cryptographic algorithms and appropriate countermeasures for securing real life applications.

COURSE OUTCOMES

After the completion of the course, the student will be able to

Course Outcomes	Description	Level
CO1	Identify the security services provided against different types of security attacks.**:**	Understand
CO2	Illustrate classical encryption techniques for information hiding.	Apply
CO3	Illustrate symmetric/asymmetric key cryptosystems for secure communication.	Apply
CO4	Explain message integrity and authentication methods in a secure communication scenario.	Understand
CO5	Interpret public/secret key distribution techniques for secure communication.	Understand
CO6	Identify the effects of intruders, malicious software and distributed denial of service attacks on system security.	Understand

SYLLABUS

Basics of Security and Traditional Cryptosystems: OSI security architecture – Security attacks, Services, Mechanisms. Cryptography vs Cryptanalysis. Classical encryption techniques – Symmetric cipher model. Substitution ciphers – Monoalphabetic vs Polyalphabetic ciphers, Caesar cipher, Affine cipher, Playfair cipher, Vigenere cipher, Hill cipher. Transposition ciphers – Keyless, Keyed, Double transposition.

Modern Symmetric Key Cryptosystems: Symmetric key ciphers – Block vs Stream ciphers, Block cipher components, Product ciphers, Feistel and Non-Feistel ciphers. Data Encryption Standard (DES) – Structure, Key generation, Design criteria, Weaknesses, Double DES, Triple DES. Advanced Encryption Standard (AES) – Structure, Key expansion. Block cipher modes of operation – Electronic Codebook Mode (ECB), Cipher Block Chaining Mode (CBC), Cipher Feedback Mode (CFB), Output Feedback Mode (OFB), Counter Mode (CTR). Stream ciphers – Structure, RC4.

Public Key Cryptosystems : Introduction to public key cryptosystems – Principles, Applications,

Requirements, Conventional vs Public key cryptosystems. RSA cryptosystem – Algorithm, Security, Attacks. ElGamal cryptosystem – Algorithm. Diffie-Hellman key exchange – Algorithm, Man-in-the-middle attack. Elliptic Curve Cryptography (ECC) – ElGamal ECC, Key exchange using ECC.

Message Integrity and Authentication: Hash functions – Security requirements, Secure Hash Algorithm (SHA-512). Message Authentication Code (MAC) – Requirements, Uses, Hash-based MAC (HMAC), Cipher-based MAC (CMAC). Digital signatures – Attacks, Forgeries, Requirements, Direct vs Arbitrated digital signatures, RSA digital signature, ElGamal digital signature, Digital Signature Standard (DSS).

Key Distribution and System Security: Key management – Distribution of secret keys using symmetric and asymmetric encryption, Distribution of public keys. System security – Intruders, Intrusion detection techniques, Password management. Malicious software – Viruses, Related threats, Countermeasures. Distributed Denial of Service (DDoS) attacks – Types, Countermeasures.

TEXT BOOKS

1. William Stallings, Cryptography and Network Security Principles and Practice, 4/e, Pearson Ed.
2. Behrouz A Forouzan, Cryptography and Network Security, 3/e, Tata McGraw-Hill.

REFERENCES

1. Charles P Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies, Security in Computing, 5/e, Prentice Hall.
2. G.A. Jones & J.M. Jones, Elementary Number Theory, Springer UTM, 2007.

COURSE PLAN

Module	Contents	No.of Hours
1	**Basics of Security and Traditional Cryptosystems** OSI security architecture – Security attacks, Services, Mechanisms, Cryptography vs Cryptanalysis. Classical encryption techniques – Symmetric cipher model, Substitution ciphers – Monoalphabetic vs Polyalphabetic ciphers, Caesar cipher, Affine cipher, Playfair cipher, Vigenere cipher, Hill cipher, Transposition ciphers – Keyless, Keyed, Double transposition	9
2	**Modern Symmetric Key Cryptosystems** Symmetric key ciphers – Block vs Stream ciphers, Block cipher components, Product ciphers, Feistel and Non-Feistel ciphers, Data Encryption Standard (DES) – Structure, Key generation, Design criteria, Weaknesses, Double DES, Triple DES, Advanced Encryption Standard (AES) – Overall Structure, Stages of encryption/decryption, Key expansion, Block cipher modes of operation – Electronic Codebook Mode (ECB), Cipher Block Chaining Mode (CBC), Cipher Feedback Mode (CFB), Output Feedback Mode (OFB), Counter Mode (CTR), Stream ciphers – Structure, RC4	9
3	**Public Key Cryptosystems** Public key cryptosystems – Principles, Applications, Requirements, Conventional vs Public key cryptosystems, RSA Security, Attacks, ElGamal cryptosystem – Algorithm, Diffie-Hellman key exchange – Algorithm, Man-in-the-middle attack, Elliptic Curve Cryptography (ECC) – ElGamal ECC, Key exchange using ECC	**9**
4	**Message Integrity and Authentication**	9

	Hash functions – Security requirements, Secure Hash Algorithm (SHA-512), Message Authentication Code (MAC) – Requirements, Uses, Hash-based MAC (HMAC), Cipher-based MAC (CMAC), Digital signatures – Attacks, Forgeries, Requirements, Direct Vs Arbitrated digital signatures, RSA digital signature, ElGamal digital signature, Digital Signature Standard (DSS)	
5	**Key Distribution and System Security** Key management – Distribution of secret keys using symmetric and asymmetric encryption, Distribution of public keys, System security – Intruders, Intrusion detection techniques, Password management, Malicious software – Viruses, Related threats, Virus countermeasures, Distributed Denial of Service (DDoS) attacks – Types, Countermeasures	**9**
	Total hours	**45**

CS1U41D	MODEL BASED SOFTWARE DEVELOPMENT	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW

The objective of the course is to familiarize learners about the concepts and advantages of using model based software development. This course covers the methodologies in developing the model of a software, perform analysis on the model and automatic generation of code from the model. The OSATE framework and its plugins using the Architecture Analysis and Design Language(AADL) language is used in the course to demonstrate the end-to-end concept of MBSD which helps the learners to get a hands on experience.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Explain the relevance of model based software development in the software development process.	Understand
CO2	Explain Model Driven Architecture with Computation Independent Model (CIM), Platform Independent Model(PIM), Platform Specific Model (PSM).	Apply
CO3	Illustrate software modeling with Architecture Analysis and Design Language (AADL).	Apply
CO4	Explain error annex using error modelling concepts and illustrate error modelling in AADL.	Understand
CO5	Illustrate the process of code generation from an AADL model.	Understand

SYLLABUS

Introduction to Model Based Software Development: Software faults, Introduction to Model checking, Introduction to Automated Testing, Model Based Software Development (MBSD) – Need, MBSD Approach, Learning MBSD from the perspective of Architecture Analysis and Design Language (AADL).

MBSD based software development: Requirements, Analysis, Design and Implementation. Model-Driven Architecture - Definitions and Assumptions, Overview of MBSD methodology, The modeling Levels-Computation Independent Model (CIM), Platform Independent Model (PIM), Platform Specific Model (PSM). Introduction to AADL, Basic Comparison of AADL with other modeling languages - Comparison with UML.

Modeling: Developing a Simple Model - Define the components - Explain with example (powerboat autopilot system), Develop a top-level model - Use Example Powerboat Autopilot (PBA) system.

AADL: Components - Software, Hardware, Composite, Runtime semantics, Language syntax, AADL declarations, AADL classifiers, AADL system models and specifications

Model Analysis: Safety Analysis -Fault tree analysis, Minimal cutsets. Error Modeling in AADL-Error

Model Libraries and Subclause Annotations, Error Types and Common Type Ontology, Error Sources and Their Impact, Component Error Behavior, Compositional Abstraction of Error Behavior, Use of Properties in Architecture Fault Models, Error modeling example.

Code Generation: Need for code generation, Categorization, Code Generation Techniques, Code Generation in AADL Model – Ocarina.

TEXT BOOKS

1. Marco, Brambilla, Jordi Cabot, Manuel Wimmer, Model-Driven Software Engineering in Practice, 2/e, Synthesis Lectures on Software Engineering, 2017.
2. Christel Baier and Joost-Pieter Katoen, Principles of model checking, The MIT Press.
3. Thomas Stahl and Markus Volter, Model-Driven Software Development, Wiley, 2006.
4. David P. Gluch, Peter H. Feiler, Model-Based Engineering with AADL: An Introduction to the SAE Architecture Analysis & Design Language, Adison-Wesley, 2015.

REFERENCES

1. Automated software testing : http://www2.latech.edu
2. Peter H. Feiler, David P. Gluch, John J. Hudak.The Architecture Analysis & Design Language(AADL): An Introduction.
3. de Niz, Dionisio, Diagrams and Languages for Model-Based Software Engineering of EmbeddedSystems: UML and AADL
4. FAA System Safety Handbook, Chapter 8: Safety Analysis/Hazard Analysis Tasks
5. Enno Ruijters, Marielle Stoelinga, Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools.
6. Larson, Brian &Hatcliff, John & Fowler, Kim &Delange, Julien. (2013). Illustrating the AADL error modeling annex (v.2) using a simple safety-critical medical device. ACM SIGAda Ada Letters. 33. 65-84. 10.1145/2527269.2527271.
7. Delange, Julien&Feiler, Peter &Hudak, John &Gluch, Dave. (2016). Architecture Fault Modeling and Analysis with the Error Model Annex, Version 2. 10.13140/RG.2.1.4224.7927.

COURSE PLAN

Module	Contents	No. of Hours
1	**Introduction** Software faults, Introduction to Model Checking, Introduction to Automated Testing (Lecture 1), Introduction to Automated Testing (Lecture 2), Need for MBSD, MBSD Approach, Architecture centric model driven software development, AADL and architecture-centric model-based software systems	9
2	**Model Based Software Development** Model based software development process, Overview of MBSD methodology, Model Driven Architecture, MDA Definitions and Assumptions, The modeling levels, Introduction to AADL, Comparison of AADL with other modeling languages	9
3	**Modeling using AADL** Modeling in detail: AADL components, Modeling in detail: Developing a simple model, Modeling in detail: Define top level model with an example, AADL in detail: Explain AADL components, Language syntax, AADL declarations and classifiers, AADL system models and specifications, Case study: Power boat auto pilot system	9
4	**Model Analysis** Introduction to safety analysis, Fault tree analysis, minimal cutsets, Error modeling with AADL - Error Model Libraries and Subclause Annotations, Error modeling with AADL - Error Types and Common Type Ontology, Error modeling with AADL - Error Sources and Their Impact, Component Error Behavior, Error modelling with AADL - Compositional Abstraction of Error Behavior, Use of Properties in Architecture Fault Models, Illustrate isolette error model	9
5	**Code Generation** Code generation and its advantages, Categorization, Code generation techniques - Templates + filtering, Template + metamodel, Frame processors, Code generation techniques - API-based generators, In-line generation, Code attributes, Code generation techniques - Code weaving Commonalities and Differences Between the Different Code generation Approaches, Code generation in AADL – Ocarina, Illustration of code generation using AADL model	9
	Total hours	**45**

CS1U41E	WEB PROGRAMMING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

This course helps the learners to understand the web programming concepts. It includes the essential frontend and backend technologies needed for the development of web applications. The learners will have an opportunity to gain necessary web development skills such as HTML, CSS, JavaScript, PHP, MySQL integration, JSON and Laravel framework.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Use HyperText Markup Language (HTML) for authoring web pages and understand the fundamentals of WWW.	Understand
CO2	Construct and visually format responsive, interactive web pages using CSS and JavaScript (JS) .	Apply
CO3	Construct websites using advanced sever side programming tool PHP.	Apply
CO4	Develop dynamic web applications using PHP and perform MySQL database operations.	Apply
CO5	Explain the importance of object exchange formats using JSON and the MVC based web application development frameworks (Laravel)	Understand

SYLLABUS

Introduction to the Internet & WWW: Evolution of Internet & World Wide Web- Web Basics, URI's & URL-MIME.

Introduction to HTML5: Structuring & editing an HTML5 document, Fundamentals of HTML -Headings-Hyper Links- Images - Special Characters & Horizontal Rules-Lists- Tables -Forms -Internal Linking- Meta Elements-HTML5 Form input types -Input and Data List Elements and autocomplete attribute- Page Structure Elements -Multimedia-HTML5 Audio & video elements.

Module -2 (CSS, JavaScript)

Introduction to Stylesheets: Introduction to CSS-Basic syntax and structure-Inline Styles, Embedded Style Sheets, Conflict Resolution, Linking External Style Sheets-Exploring CSS Selectors-Properties, values, Positioning Elements: Absolute Positioning, Relative Positioning Backgrounds-List Styles-Element Dimensions- Table Layouts-Box Model and Text Flow-div and span -Basics of Responsive CSS, Media port & Media Queries.

Introduction to JavaScript : Introduction to Scripting- Programming fundamentals of JavaScript -Obtaining User Input with prompt Dialogs-Arithmetic-Decision Making -Control Statements - Functions -Arrays -Objects -Document Object Model (DOM) -Form processing

PHP Language Structure: Introduction- **B**uilding blocks of PHP-Variables, Data Types - simple PHP program-Converting between Data Types- Operators and Expressions -Flow Control functions - Control statements- Working with Functions- Initialising and Manipulating Arrays--Objects- String Comparisons-String processing with Regular Expression

Advanced PHP: Form processing and Business Logic-Cookies- Sessions & MySQL Integration-Connecting to MySQL with PHP- Performing CREATE, DELETE, INSERT, SELECT and UPDATE operations on MySQL table -Working with MySQL data-Reading from Database-Dynamic Content.

JSON Data Interchange Format: Syntax, Data Types, Object, JSON Schema, Manipulating JSON data with PHP

Web Development Frameworks: Laravel Overview-Features of Laravel-Setting up a Laravel Development Environment-Application structure of Laravel-Routing -Middleware-Controllers-Route Model Binding-Views-Redirections-Request and Responses.

TEXT BOOKS

1 Paul J. Deitel, Harvey M. Deitel, Abbey Deitel, Internet & World Wide Web How to Program 5th Edition **[Module 1,2,3,4]**

2. Lindsay Bassett, Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON 1st Edition, O'Reilly **[Module 5]**
3. Julie C. Meloni, Pearson -PHP, MySQL & JavaScript All in One, Sams Teach Yourself,5th Ed [**Module 4]**
4. Matt Stauffer," LARAVEL up and Running, A framework for building modern PHP apps"1st Edition, O'REILLY **[Module 5]**

REFERENCES

1. Robert W Sebesta, Programming the World Wide Web, 7/e, Pearson Education Inc,8th Edition
2. Larry Ullman, Pearson- PHP 6 and MySQL 5 for Dynamic Web Sites: Visual QuickPro Guide
3. Eric van der Vlist, Danny Ayers, Erik Bruchez, Joe Fawcett, Alessandro Vernet",Wrox-Professional Web 2.0 Programming, Wiley-India edition
4. Web Technologies Black Book 2018(As per Mumbai University Syllabus) HTML, CSS3, JavaScript, iQuery, AJAX,PHP,XML,MVC and Laravel DT Editorial Services (ISBN: 9789386052490)

COURSE PLAN

Module	Contents	No. of Hours
1	**Introduction to Internet and WWW** Evolution of Internet &World Wide Web- Web Basics URI's & URL -MIME. **Introduction to HTML5** Structuring & editing an HTML5 document- Fundamentals of HTML, Headings-Images, Hyper Links, Internal Linking- Lists, Special Characters & Horizontal Rules- meta Elements- div and span, Tables- Forms,HTML5 Form input types, input and data list Elements and autocomplete attributes-Page Structure Elements, Multimedia-HTML5 Audio & video elements.	9

2	**Introduction to Cascading Style Sheets(CSS)** Introduction to CSS3-Basic syntax and structure-Inline Styles ,Embedded Style Sheets-Linking External Style Sheets,Exploring CSS Selectors-Properties-values,Positioning Elements: Absolute Positioning- Relative Positioning -Backgrounds-List Styles- Table Layouts,Box Model and Text Flow, Basics of Responsive CSS-Media port & Media Queries **Introduction to JavaScript** Introduction to Scripting- Programming fundamentals of JavaScript -Obtaining User Input with prompt Dialogs, Arithmetic-Decision Making Control Statements -Functions,Arrays - Objects,Document Object Model (DOM)- Form processing	9
3	**Introduction to PHP** Building blocks of PHP-Variables, Data Types simple PHP program , Converting between Data Types, Operators and Expressions -Flow Control functions , Control Statements -Working with Functions, Initialising and Manipulating Arrays- Objects,Working with Strings-String processing with Regular expression, Pattern Matching , Form processing and Business Logic.	9
4	**PHP –MYSQL** Cookies- Sessions, PHP& MySQL Integration-Connecting to MySQL with PHP , Working with MySQL data , Performing CREATE, DELETE, INSERT operations on MySQL table from PHP Program., Performing SELECT and UPDATE operations on MySQL table from PHP Program, Building Dynamic Content in PHP application	9
5	**JSON** JSON Data Interchange Format -Syntax, Data Types, Object JSON Schema, Manipulating JSON data with PHP **LARAVEL** Laravel Overview- Design Pattern- Laravel Feature, Setting up a Laravel Development Environment-Application structure of Laravel, Laravel Basics Routing -middleware - Controllers, Route Model Binding-Views-Redirections, Blade Templating-echoing data, control structures	9
	Total hours	**45**

CS1U41F	NATURAL LANGUAGE PROCESSING	CATEGORY	L	T	P	CREDIT	YEAROF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW

This course enables the learners to understand the concepts of Natural Language Processing. The course covers basic pre-processing steps, language models, text classification using machine learning algorithms, information and relation extraction methods, Information Retrieval, Question Answer Systems and Machine Translation models. This course enables the students to apply techniques and methods to solve challenging real-world problems in NLP.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Summarize basic concepts and learning methods for NLP	Understand
CO2	Demonstrate the relevance of pre-processing methods on text data.	Apply
CO3	Compare different language modelling techniques.	Apply
CO4	Make use of NLP techniques in Text Classification and Information Retrieval	Apply
CO5	Explain Information Extraction, Relation Detection, QA Systems and Machine Translation**.**	Understand

SYLLABUS

Introduction to NLP: NLP Tasks and Applications, Language-Building Blocks, Challenges of NLP, Machine Learning for NLP – Naïve Bayes Classifier, Logistic Regression, Support Vector Machines, Approaches to NLP-- Heuristics-Based NLP, Machine Learning-based NLP.

Pre-processing and Representation Models: NLP System Pipeline--Steps--Data Acquisition, Text Extraction and Clean-up, Pre-processing, Feature Engineering, Modelling, Evaluation, Post-Modelling Phases.Text Representation--Vector Space Models--Basic Vectorization Approaches--One-Hot Encoding, Bag of Words, Bag of N-Grams TF-IDF; Distributed Representations-- Word Embeddings, Doc2Vec.

Classification and Information Extraction: Text Classification--Text classification applications – Pipeline for building text classification systems, Naïve Bayes for Sentiment Classification – Naïve Bayes Classifier Training –Optimizing for Sentiment Analysis, Logistic Regression, Support Vector Machine for Text Classification.Information Extraction(IE)—IE Applications – The General Pipeline for IE - Named Entity Recognition(NER), Ambiguity in Named Entity Recognition – NER as Sequence Labeling –Evaluation of NER.

Relation Detection and Information Retrieval: Relation Detection and Classification – Supervised Learning Approaches to Relation Analysis –Lightly Supervised Approaches to Relation Analysis – Evaluation of Relation Analysis systems.Information Retrieval – Term weighting and document scoring – Inverted Index – Evaluation of Information Retrieval Systems.

QA Systems and Machine Translation: Question-Answering Systems – Factoid Question Answering – Question Processing – Passage Retrieval – Answer Processing – Evaluation of Factoid Answers.Machine Translation – Why Machine Translation is Hard – Classical Machine Translation – Direct Translation – Transfer – Statistical Machine Translation- The Phrase based Translation model – Alignment in MT – Training Alignment Models – Symmetrizing Alignments for Phrase-based MT – Decoding for Phrase-based Statistical MT

TEXT BOOKS

1. Daniel Jurafsky, James H. Martin , “Speech and Language Processing”(2nd and 3rd editions), Pearson Prentice Hall
2. Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, Harshit Surana,” Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems “ June 2020 Publisher(s): O'Reilly Media, Inc. ISBN: 9781492054054.

REFERENCES

1. James Allen, “Natural Language Understanding”, Second Edn , Pearson.
2. Christopher Manning and Hinrich Schutze, Statistical Natural Language Processing, MIT Press.

COURSE PLAN

Module	Contents	No.of Hours
1	**Introduction to NLP** Introduction to NLP – Tasks and Applications, Language – Building Blocks, Challenges of NLP, Approaches to NLP - Heuristics-Based NLP, Machine Learning for NLP, Machine Learning for NLP – Naïve Bayes Classifier, Logistic Regression, Support Vector Machines – Linearly Separable Data, Support Vector Machines – Linearly Inseparable Data	**9**

2	**Pre-processing and Representation Models** NLP System Pipeline – Stages – Overview, Data Acquisition, NLP System Pipeline – Text Extraction and Cleanup, NLP System Pipeline – Preprocessing - Sentence segmentation, Word tokenization, Stemming and lemmatization, Feature Engineering, Model Building, Evaluation – Metrices, Post-modeling phase, Text Representation – Vector Space Model, Vectorization Approaches – One hot encoding, Bag of words, Bag of n-grams, TF-IDF, Word Embeddings – Word2Vec- CBOW, SkipGram models	9
3	**Classification and Information Extraction** Text Classification--Text classification applications -Pipeline for building text classification systems, Sentiment Analysis using Naïve Bayes Classifier, Case Studies for Text Classification using Logistic Regression and Support Vector Machines, Information Extraction (IE) and Applications, IE Tasks and the IE Pipeline, Named Entity Recognition (NER) – Ambiguity in NER, NER as Sequence Labeling, Evaluation of NER, Practical NER Systems	9
4	**Relation Detection and Information Retrieval** Relation Detection and Classification – Supervised Learning Approaches to Relation Analysis, Relation Detection and Classification – Lightly Supervised Approaches to Relation Analysis, Relation Detection and Classification -Evaluation of Relation Analysis systems, Information Retrieval – Term weighting and document scoring, Inverted Index, Evaluation of Information-Retrieval Systems	9
5	**QA Systems and Machine Translation** Question-Answering Systems – Factoid Question Answering, Question Processing, Passage Retrieval, Answer Processing, Evaluation of Factoid Answers, Machine Translation – Why Machine Translation is Hard, Classical Machine Translation , Statistical Machine Translation, The Phrase based Translation model, Alignment in Machine Translation, Decoding for Phrase-based Statistical MT	9
	Total hours	**45**

OPEN ELECTIVE

CS0U41	INTRODUCTION TO MOBILE COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		OEC	2	1	0	3	2020

COURSE OVERVIEW

The purpose of this course is to prepare learners to understand the functionalities and design considerations of mobile computing. The course content is designed to cover the mobile computing architecture, features of different communication systems and major elements of mobile security and next generation computer systems. This course enables the learners to acquire advanced concepts on mobile and ad-hoc networks.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Describe the mobile computing applications, services, design considerations and architectures	Understand
CO2	Identify the technology trends for cellular wireless networks **.**	Understand
CO3	Summarize the Short Messaging Service and General Packet Radio Service	Understand
CO4	Outline the LAN technologies used in mobile communication.	Understand
CO5	Describe the security protocols and apply suitable security algorithm to secure the communication .	Apply
CO6	Explain the fundamental concepts of next generation mobile networks**.**	Understand

SYLLABUS

Mobile Computing Architecture: Introduction to mobile computing – Functions, Devices, Middleware and gateways, Applications and services, Limitations. Mobile computing architecture – Internet: The ubiquitous network, Three-tier architecture, Design considerations for mobile computing.

Communication Systems: Mobile computing through telephony - Evolution of telephony, Multiple access procedures - Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), Space Division Multiple Access (SDMA). Satellite communication systems – Basics, Applications, Geostationary Earth Orbit (GEO), Low Earth Orbit (LEO), Medium Earth Orbit (MEO), Satellite phones. Mobile computing through telephone – Interactive Voice Response (IVR) architecture, Overview of voice software, Developing an IVR application. Global System for Mobile Communication (GSM) - Introduction, Architecture, Entities, Call routing, Mobility management, Frequency allocation, Authentication and security.

Short Messaging Service and General Packet Radio Service: Short Message Service (SMS) – Strengths, Architecture, Value added services, Accessing the SMS bearer. General Packet Radio Service (GPRS) – Architecture, Network operations, Data services, Applications, Limitations, Billing and charging.

Wireless Local Area Networks: Wireless Local Area Network (WLAN) - Advantages, Evolution, Applications, Architecture, Mobility, Security, Deploying WLAN. Wireless Local Loop (WLL) – Architecture. High Performance Radio Local Area Network (HIPERLAN). WiFi Vs 3G.

Mobile Security and Next Generation Networks: Security issues in mobile computing - Information security, Security techniques and algorithms, Security protocols. Next generation networks – The Converged Scenario, Narrowband to broadband, Orthogonal Frequency Division Multiplexing (OFDM), Multi Protocol Label Switching (MPLS), Wireless Asynchronous Transfer Mode (WATM), Multimedia broadcast services.

TEXT BOOKS

1. Asoke K. Talukder, Hasan Ahmad, Roopa R Yavagal, Mobile Computing Technology- Application and Service Creation, 2nd Edition, McGraw Hill Education.
2. Schiller J., Mobile Communications, 2/e, Pearson Education, 2009

REFERENCES

1. Andrew S. Tanenbaum, Computer Networks, 6/e, PHI.
2. Theodore S. Rappaport, Wireless Communications Principles and Practice, 2/e, PHI, New Delhi, 2004.
3. Curt M. White, Fundamentals of Networking and Communication 7/e, Cengage learning.

COURSE PLAN

Module	Contents	No.of Hours
1	**Mobile Computing Architecture** Introduction to mobile computing – Functions, Devices, Middleware and gateways, Applications, services, limitations, Internet: The ubiquitous network, Three-tier architecture (Lecture 1), Three-tier architecture (Lecture 2), Design considerations for	9

	mobile computing (Lecture 1), Design considerations for mobile computing (Lecture 2)	
2	**Communication Systems** Evolution of telephony, Multiple access procedures –FDMA, TDMA, CDMA, SDMA, Satellite communication systems – GEO, MEO, LEO, Satellite phones, Interactive Voice Response (IVR) architecture, Overview of voice software, Developing an IVR application (Call flow diagram), Introduction to GSM,Architecture, GSM entities, Call routing, Mobility management, Frequency allocation, Authentication and security	9
3	**Short Messaging Service and General Packet Radio Service** SMS Strengths, Architecture, Short Message Mobile Terminated (SM MT) and Short Message Mobile Originated (SM MO) messages, SMS Architecture - Operator-centric pull, operator-independent push/pull, Value added services, Accessing the SMS bearer (Lecture 1), Accessing the SMS bearer (Lecture 2), GPRS architecture, Network operations Data services, Applications,Limitations, Billing and charging	9
4	**Wireless Local Area Networks** WLAN Advantages, Evolution, Applications, WLAN Architecture (Lecture 1), WLAN Architecture (Lecture 2) Mobility, Security, Deploying WLAN, WLL Architecture, HIPERLAN, WiFi Vs 3G	9
5	**Mobile Security and Next Generation Networks** Information security – Attacks, Components, Security techniques and algorithms – Stream Vs Block cipher, Symmetric Vs Asymmetric cryptography, Security techniques and algorithms – RSA, Diffie Hellman Key exchange, Security protocols – Secure Socket Layer, Transport Layer Security, Wireless Transport Layer Security, The Converged Scenario, Narrowband to broadband, Orthogonal Frequency Division Multiplexing (OFDM) and Multi Protocol Label Switching (MPLS), Wireless Asynchronous Transfer Mode (WATM) and Multimedia broadcast services	9
	Total hours	**45**

CS0U41B	INTRODUCTION TO DEEP LEARNING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		OEC	2	1	0	3	2020

COURSE OVERVIEW

This course aims to introduce the learner to an overview of the concepts and algorithms involved in deep learning. Basic concepts and application areas of machine learning, deep networks, convolutional neural network and recurrent neural network are covered in this course. This is a foundational program that will help students understand the capabilities, challenges, and consequences of deep learning and prepare them to participate in the development of leading-edge AI technology. They will be able to gain the knowledge needed to take a definitive step in the world of AI.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Demonstrate basic concepts in machine learning.	Understand
CO2	Illustrate the validation process of machine learning models using hyper-parameters and validation sets.	Understand
CO3	Demonstrate the concept of the feed forward neural network and its training process.	Apply
CO4	Build CNN and Recurrent Neural Network (RNN) models for different use cases.	Apply
CO5	Use different neural network/deep learning models for practical applications.	Apply

SYLLABUS

Introduction: Key components - Data, models, objective functions, optimization algorithms, Learning algorithms. Supervised learning- regression, classification, tagging, web search, page ranking, recommender systems, sequence learning, Unsupervised learning, Reinforcement learning, Historical Trends in Deep Learning. Other Concepts - overfitting, underfitting , hyperparameters and validation sets, estimators, bias and variance.

Optimization and Neural Networks: Neural Networks –Perceptron, Gradient Descent solution for Perceptron, Multilayer perceptron, activation functions, architecture design, chain rule, back propagation, gradient based learning. Introduction to optimization– Gradient based optimization, linear least squares. Stochastic gradient descent, Building ML algorithms and challenges.

Convolutional Neural Network: Convolutional Neural Networks – convolution operation, motivation, pooling, Convolution and Pooling as an infinitely strong prior, variants of convolution functions, structured outputs, data types, efficient convolution algorithms.

Recurrent Neural Network: Recurrent neural networks – Computational graphs, RNN design, encoder – decoder sequence to sequence architectures, deep recurrent networks, recursive neural networks, modern RNNs LSTM and GRU, Practical use cases for RNNs.

Application Areas: Applications – computer vision, speech recognition, natural language processing. Research Areas – Autoencoders, Representation learning, Boltzmann Machines, Deep belief networks.

TEXT BOOKS

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press 2015 ed.
2. Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola, Dive into Deep Learning, August 2019.
3. Neural Networks and Deep Learning: A Textbook by Charu C. Aggarwal. Springer.1st edition, 2018.

REFERENCE BOOKS

1. Neural Smithing: Supervised Learning in Feed forward Artificial Neural Networks by Russell Reed, Robert J MarksII, 1st edition, 1999, MIT Press.
2. Practical Convolutional Neural Networks by Mohit Sewak, Md. Rezaul Karim, Pradeep Pujari, 1st edition, 2018, Packt Publishing Ltd.
3. Hands-On Deep Learning Algorithms with Python by Sudharsan Ravichandran, 1st edition, 2019, Packt Publishing Ltd.
4. Deep Learning with Python by Francois Chollet, 2nd edition, 2018, Manning Publications

COURSE PLAN

Module	Contents	No.of Hours
1	**Introduction** Key components - Data, models, objective functions, optimization algorithms, Learning algorithm, Supervised learning- regression, classification, tagging, web search, page ranking ,Recommender systems, Sequence learning, Unsupervised learning, Reinforcement learning, Historical Trends in Deep Learning. Concepts: overfit, underfit, hyperparameters and validation sets., Concepts: Estimators, bias and variance., Demonstrate the concepts of supervised learning algorithms using a suitable platform, Demonstrate the concepts of unsupervised using a suitable platform.	9
2	**Optimization and Neural Networks** Perceptron, Stochastic Gradient descent, Gradient descent solution for perceptron ,Multilayer perceptron, , Activation functions- Sigmoid, tanh, Softmax, ReLU, leaky ReLU Architecture design, Chain rule, back propagation,Gradient based learning, Gradient based optimization, Linear least squares using a suitable platform. Building ML Algorithms and Challenges	9
3	Convolution Neural Network Convolution operation, Motivation, pooling ,Convolution and Pooling as an infinitely strong prior ,Variants of convolution functions – multilayer convolutional network, tensors, kernel flipping, downsampling, strides and zero padding. Variants of convolution functions - unshared convolutions, tiled convolution, training different networks.,Structured outputs, data types, Efficient convolution algorithms., Case Study: AlexNet, VGG, ResNet.	9
4	Recurrent Neural Network Computational graphs, RNN ,Encoder – decoder sequence to sequence	9

	architectures. Deep recurrent networks .Recursive neural networks , Modern RNNs, LSTM and GRU, Practical use cases for RNNs, Demonstrate the concepts of RNN using a suitable platform.	
5	**Applications and Research** Computer vision. Speech recognition, Natural language processing., Brief introduction on current research areas- Autoencoders, Representation learning. Brief introduction on current research areas- Boltzmann Machines, Deep belief networks.	9
	Total hours	**45**

CS0U41C	COMPUTER GRAPHICS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		OEC	2	1	0	3	2020

COURSE OVERVIEW

This course helps the learners to make awareness about strong theoretical concept in computer graphics. It covers the three-dimensional environment representation in a computer, transformation of 2D/3D objects and basic mathematical techniques and algorithms used to build applications. This course enables the learners to develop the ability to create image processing frameworks for different domains and develop algorithms for emerging display technologies.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Describe the working principles of graphics devices**.**	Understand
CO2	Illustrate line drawing, circle drawing and polygon filling algorithms**.**	Apply
CO3	Demonstrate geometric representations and transformations on 2D & 3D objects	Apply
CO4	Demonstrate the working of line and polygon clipping algorithms	Apply
CO5	Summarize visible surface detection methods and illustrate projection algorithms.	Apply

SYLLABUS

Basics of Computer Graphics: Basics of Computer Graphics and its applications. Video Display devices- Refresh Cathode Ray Tubes(CRT), Random Scan Displays and systems, Raster scan displays and systems, Color CRT displays, Flat panel display and its categories.

Line drawing, Circle drawing and Filled Area Primitives: Line drawing algorithms- DDA, Bresenham's algorithm. Circle drawing algorithms- Midpoint Circle generation algorithm, Bresenham's algorithm. Filled Area Primitives- Scan line polygon filling, Boundary filling and flood filling.

Geometric transformations: Two dimensional transformations -Translation, Rotation, Scaling, Reflection and Shearing, Composite transformations, Matrix representations and homogeneous coordinates. Basic 3D transformations.

Clipping: Window to viewport transformation. Cohen Sutherland and Midpoint subdivision line clipping algorithms, Sutherland Hodgeman and Weiler Atherton Polygon clipping algorithms.

Three dimensional graphics: Three dimensional viewing pipeline. Projections- Parallel and Perspective projections. Visible surface detection algorithms- Back face detection, Depth buffer algorithm, Scan line algorithm, A buffer algorithm

TEXT BOOKS

1. Zhigang Xiang and Roy Plastock, Computer Graphics (Schaum's outline Series), McGraw Hill, 2019.
2. Donald Hearn and M. Pauline Baker, Computer Graphics, PHI, 2e, 1996

REFERENCES

1. William M. Newman and Robert F. Sproull, Principles of Interactive Computer Graphics. McGraw Hill, 2001
2. David F. Rogers , Procedural Elements for Computer Graphics, Tata McGraw Hill,2001.
3. Donald Hearn, M. Pauline Baker and Warren Carithers, Computer Graphics with OpenGL, PHI, 4e, 2013

COURSE PLAN

Module	Contents	No.of Hours
1	**Basics of Computer Graphics** Basics of Computer Graphics and applications, Refresh Cathode Ray Tubes, Random Scan Displays and systems, Raster scan displays and systems, Color CRT displays Flat panel display and its categories.	9
2	**Line drawing, Circle drawing and Filled Area Primitives** DDA Line drawing Algorithm, Bresenham's line drawing algorithm, Midpoint Circle generation algorithm, Bresenham's Circle generation algorithm, Illustration of line drawing and circle drawing algorithms, Scan line polygon filling, Boundary filling and flood filling	9
3	**Geometric transformations** Basic 2D transformations-Translation and Rotation, Basic 2D transformations- Scaling, Reflection and Shearing, Illustration of 2D Transformations, Composite transformations, Matrix representations and homogeneous coordinates, Basic 3D transformations, Illustration of basic 3D transformations	9
4	**2D Clipping** Window to viewport transformation, Cohen Sutherland Line clipping	9

	algorithm, Midpoint subdivision Line clipping algorithm Sutherland Hodgeman Polygon clipping algorithm, Weiler Atherton Polygon clipping algorithm, Practice problems on Clipping algorithms	
5	**Three dimensional graphics** Three dimensional viewing pipeline, Projections-Parallel projections, Projections- Perspective projections, Visible surface detection algorithms-Back face detection., Depth buffer algorithm, Depth buffer algorithm, Scan line visible surface detection algorithm, Scan line visible surface detection algorithm, A buffer algorithm	9
	Total hours	**45**

CS0U41D	PYTHON FOR ENGINEERS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		OEC	2	1	0	3	2020

COURSE OVERVIEW

The objective of the course is to provide learners an insight into Python programming in a scientific computation context and develop programming skills to solve engineering problems. It covers programming environment, important instructions, data representations, intermediate level features, Object Oriented Programming and file data processing of Python. This course lays the foundation to scientific computing, develop web applications, Machine Learning, and Artificial Intelligence-based applications and tools, Data Science and Data Visualization applications.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Write, test and debug Python programs	Apply
CO2	Illustrate uses of conditional (if, if else, if-elif-else and switch-case) and iterative (while and for) statements in Python programs .	Apply
CO3	Develop programs by utilizing the modules Lists, Tuples, Sets and Dictionaries in Python.	Apply
CO4	Implement Object Oriented programs with exception handling **.**	Apply
CO5	Analyze, Interpret, and Visualize data according to the target application .	Apply
CO6	Develop programs in Python to process data stored in files by utilizing the modules Numpy, Matplotlib, and Pandas.	Apply

SYLLABUS

Basics of Python: Getting Started with Python Programming - Running code in the interactive shell, Editing, Saving, and Running a script. Using editors - IDLE, Jupyter. Basic coding skills - Working with data types, Numeric data types and Character sets, Keywords, Variables and Assignment statement, Operators, Expressions, working with numeric data, Type conversions, Comments in the program, Input Processing, and Output, Formatting output. How Python works. Detecting and correcting syntax errors. Using built in functions and modules in math module. Control statements - Selection structure - if-else, if-elif-else. Iteration structure - for, while. Testing the control statements. Lazy evaluation.

Functions and Python Data Structures: Functions - Hiding redundancy and complexity, Arguments and return values, Variable scopes and parameter passing, named arguments, Main function, Working with recursion, Lambda functions. Strings - String function. Lists - Basic list Operations and functions, List of lists, Slicing, Searching and sorting list, List comprehension. Work with tuples. Sets. Dictionaries -Dictionary functions, dictionary literals, adding and removing keys, accessing and replacing values, traversing dictionaries, reverse lookup.

Object Oriented Programming: Design with classes - Objects and Classes, Methods, Instance Variables, Constructor, Accessors and Mutators. Structuring classes with Inheritance and Polymorphism. Abstract Classes. Exceptions - Handle a single exception, Handle multiple exceptions.

Visualization and File handling: Plotting - An Interactive Session with PyPlot, Basic Plotting, Logarithmic Plots, More Advanced Graphical Output, Plots with multiple axes, Mathematics and Greek symbols, The Structure of matplotlib, Contour and Vector Field Plots. File Processing - The os and sys modules, Introduction to file I/O, Reading and writing text files, Working with CSV files.

Scientific Computing: Numerical Routines. SciPy and NumPy - Basics, Creating arrays, Arithmetic, Slicing, Matrix Operations, Special Functions, Random Numbers, Linear Algebra, Solving Nonlinear Equations, Numerical Integration, Solving ODEs. Data Manipulation and Analysis – Pandas : Reading Data from Files Using Pandas, Data Structures: Series and DataFrame, Extracting Information from a DataFrame, Grouping and Aggregation.

TEXT BOOKS

1. Kenneth A Lambert., Fundamentals of Python : First Programs, 2/e, Cengage Publishing, 2016
2. David J. Pine, Introduction to Python for Science and Engineering, CRC Press, 2021

REFERENCES

1. Wes McKinney, Python for Data Analysis, 2/e, Shroff / O'Reilly Publishers, 2017
2. Allen B. Downey, Think Python: How to Think Like a Computer Scientist, 2/e, Schroff, 2016
3. Michael Urban and Joel Murach, Python Programming, Shroff/Murach, 2016
4. David M.Baezly, Python Essential Reference. Addison-Wesley Professional; 4/e, 2009.
5. Charles Severance. Python for Informatics: Exploring Information,
6. http://swcarpentry.github.io/python-novice-gapminder/

COURSE PLAN

Module	Contents	No.of Hours
1	**Basics of Python**	9

	Getting Started with Python Programming: Running code in the interactive shell Editing, Saving, and Running a script, Using editors: IDLE, Jupyter, Basic coding skills: Working with data types, Numeric data types and Character sets, Keywords, Variables and Assignment statement, Operators, Expressions, Working with numeric data, Type conversions, Comments in the program, Input Processing, and Output. Formatting output, How Python works. Detecting and correcting syntax errors. Using built in functions and modules in math module. Control statements : Selection structure, if-else, if elif else, Iteration structure - for, while Testing the control statements, Lazy evaluation.	
2	**Functions and Python Data Structures** Functions: Hiding redundancy and complexity, Arguments and return values, Variable scopes and parameter passing, Named arguments, Main function, Working with recursion, Lambda functions, Strings - String function, Lists - Basic list Operations and functions, List of lists, Slicing, Searching and sorting list, List comprehension, Work with tuples. Sets, Dictionaries - Dictionary functions, dictionary literals, adding and removing keys, Accessing and replacing values, traversing dictionaries, reverse lookup	9
3	**Object Oriented Programming** Design with classes : Objects and Classes, Methods, Instance Variables Constructor, Accessors, and Mutators, Structuring classes with Inheritance, Polymorphism, Abstract Classes, Exceptions: Handle a single exception, Handle multiple exception	9
4	**Visualization and File handling** Plotting - An Interactive Session with PyPlot, Basic Plotting, Logarithmic Plots, More Advanced Graphical Output, Plots with multiple axes, Mathematics and Greek symbols, The Structure of matplotlib, Contour and Vector Field Plots File Processing -The os and sys modules, Introduction to file I/O, Reading and writing text files,Working with CSV files	9
5	**Scientific Computing** Numerical Routines: SciPy and NumPy - Basics, Creating arrays, Arithmetic, Slicing, Matrix Operations, Special Functions, Random Numbers, Linear Algebra, Solving Nonlinear Equations Numerical Integration, Solving ODEs, Data Manipulation and Analysis: Pandas - Reading Data from Files Using Pandas, Data Structures - Series and DataFrame, Extracting Information from a DataFrame, Grouping and Aggregation	9
	Total hours	**45**

CS0U41E	OBJECT ORIENTED CONCEPTS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		OEC	2	1	0	3	2020

COURSE OVERVIEW

The purpose of this course is to enable learners to solve problems by breaking it down to object level while designing software and to implement it using Java. This course covers Object Oriented Principles, Object Oriented Programming in Java, Exception handling, Event handling, multithreaded programming and working with window-based graphics. This course provides learners the basics to develop Mobile applications, Enterprise Applications, Scientific Applications and Web based Applications.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Develop Java programs using the object-oriented concepts - classes, objects, constructors, data hiding, inheritance and polymorphism.	Apply
CO2	Utilize data types, operators, control statements, built in packages & interfaces, Input/output Streams and Files in Java to develop programs **.**	Apply
CO3	Illustrate how robust programs can be written in Java using exception handling mechanism	Apply
CO4	Develop application programs in Java using multithreading **.**	Apply
CO5	Develop Graphical User Interface based application programs by utilizing event handling features and Swing in Java	Apply

SYLLABUS

Object Orientation and Java basics: Object Orientation Principles – Object and Class, Data abstraction and Encapsulation, Inheritance, Polymorphism, Dynamic binding, Message communication, Benefits of using Object orientation.Introduction to Java - Java programming Environment and Runtime Environment, Development Platforms -Standard, Enterprise. Java Virtual Machine (JVM), Java compiler, Bytecode, Java applet, Java Buzzwords, Java program structure, Comments, Garbage Collection, Lexical Issues.Primitive Data types - Integers, Floating Point Types, Characters, Boolean. Literals, Type Conversion and Casting, Variables, Arrays, Strings, Vector class.

Core Java Fundamentals: Operators - Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, Assignment Operator, Conditional (Ternary) Operator, Operator Precedence.

Control Statements - Selection Statements, Iteration Statements and Jump Statements. Object Oriented Programming in Java - Class Fundamentals, Declaring Objects, Object Reference, Introduction to Methods, Constructors, ***this*** Keyword, Method Overloading, Using Objects as Parameters, Returning Objects, Recursion, Access Control, Static Members, Command-Line Arguments, Variable Length Arguments.

More features of Java: Inheritance - Super Class, Sub Class, The Keyword ***super***, protected Members, Calling Order of Constructors, Method Overriding, the Object class, Abstract Classes and Methods, Using ***final*** with Inheritance.Packages and Interfaces - Defining Package, CLASSPATH, Access Protection, Importing Packages, Interfaces.Exception Handling - Checked Exceptions, Unchecked Exceptions, ***try*** Block and ***catch*** Clause, Multiple ***catch*** Clauses, Nested ***try*** Statements, ***throw***, ***throws*** and ***finally***.

Advanced features of Java: Input/Output - I/O Basics, Reading Console Input, Writing Console Output, PrintWriter Class, Reading and Writing Files.Java Library - String Handling – String Constructors, String Length, Special String Operations -Character Extraction, String Comparison, Searching Strings, Modifying Strings, Using valueOf(), Comparison of String Buffer and String.

GUI Programming, Event Handling and Multithreaded Programming: Multithreaded Programming - The Java Thread Model, The Main Thread, Creating Thread, Creating Multiple Threads, Suspending, Resuming and Stopping Threads.Event Handling - Event Handling Mechanisms, Delegation Event Model, Event Classes, Sources of Events, Event Listener Interfaces, Using the Delegation Model.Swing Fundamentals - Swing Key Features, Model View Controller (MVC), Swing Controls, Components and Containers, Exploring Swing - JFrame, JLabel, JButton, JTextField.

TEXT BOOKS

1. Schildt, Java: The Complete Reference, 8/e, Tata McGraw Hill, 2011.
2. Balagurusamy E., Programming JAVA a Primer, 5/e, McGraw Hill, 2014.

REFERENCES

1. Paul Deitel, Harvey Deitel, Java How to Program, Early Objects 11/e, Pearson, 2018.
2. Y. Daniel Liang, Introduction to Java Programming, 7/e, Pearson, 2013.

3. Nageswararao R., Core Java: An Integrated Approach, Dreamtech Press, 2008.
4. Flanagan D., Java in A Nutshell, 5/e, O'Reilly, 2005.
5. Sierra K., Head First Java, 2/e, O'Reilly, 2005.

COURSE PLAN

Module	Contents	No.of Hours
1	**Object Orientation and Java basics** Object Orientation Principles – Object and Class, Data abstraction and Encapsulation, Inheritance, Polymorphism, Dynamic binding, Message communication, Benefits of using Object orientation, Java programming Environment and Runtime Environment, Development Platforms -Standard, Enterprise. JVM, Java compiler, Bytecode, Java applet, Java Buzzwords, Java program structure, Comments, Garbage Collection, Lexical Issues, Primitive Data types - Integers, Floating Point Types, Characters, Boolean, Literals, Type Conversion and Casting, Variables, Arrays, Strings, Vector class	9
2	**Core Java Fundamentals** Operators - Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, Assignment Operator, Conditional (Ternary) Operator, Operator Precedence, Control Statements - Selection Statements, Iteration Statements and Jump Statements, Object Oriented Programming in Java - Class Fundamentals, Declaring Objects, Object Reference, Introduction to Methods, Constructors, this Keyword, Method Overloading, Using Objects as Parameters, Returning Objects, Recursion, Access Control, static Members, Command-Line Arguments, Variable Length Arguments	9
3	**More features of Java** Inheritance - Super class, Sub class, the keyword super, protected Members, Calling Order of Constructors, Method Overriding, the Object class, Abstract Classes and Methods, Using final with Inheritance, Packages and Interfaces - Defining Package, CLASSPATH, Access Protectio, Importing Packages, Interfaces, Exception Handling - Checked Exceptions, Unchecked Exceptions, try Block and catch Clause, Multiple catch Clauses, Nested try Statements, throw, throws and finally	9
4	**Advanced features of Java** Input/Output - I/O Basics, Reading Console Input, Writing Console Output, PrintWriter Class, Working with Files (Lecture-1), Working with Files (Lecture-2), Java Library - String Handling – String Constructors, String Length, Special String Operations - Character Extraction, String Comparison, Searching Strings, Modifying Strings, Using valueOf( ), Comparison of StringBuffer and String.	9
5	**GUI Programming, Event Handling and Multithreaded Programming** Multithreaded Programming - The Java Thread Model, The Main Thread, Creating Thread, Creating Multiple Threads, Suspending, Resuming and Stopping Threads, Event handling - Event Handling Mechanisms, Delegation Event Model, Event Classes,Sources of Events, Event Listener Interfaces, Using the Delegation Model, Swing fundamentals, Swing Key Features, Model View Controller (MVC), Swing Controls, Components and Containers, Exploring Swing –JFrame, JLabel, JButton, JTextField	9
	Total hours	**45**

MINOR

CS0M49A	MINI PROJECT	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PWS	0	1	6	4	2020

COURSE OVERVIEW

The objective of this course is to apply the fundamental concepts of different courses learned in respective Minor Streams: Software Engineering, Machine Learning and Networking. This course helps the learners to get an exposure to the development of application software/hardware solutions/ software simulations in the field of Computer Science and Engineering. It enables the learners to understand the different steps to be followed such as literature review and problem identification, preparation of requirement specification & design document, testing, development and deployment. Mini project enables the students to boost their skills, widen the horizon of thinking and their ability to resolve real life problems.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Identify technically and economically feasible problems	Apply
CO2	Identify and survey the relevant literature for getting exposed to related solutions.	Apply
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques	Apply
CO4	Prepare technical report and deliver presentation	Apply
CO5	Apply engineering and management principles to achieve the goal of the project	Apply

COURSE PLAN

Students are expected to follow the following steps.

1. Review of Literature and Identification of a problem
2. Create an abstract with a problem statement, solution approach, technology stack, team, etc.
3. Create Requirements Specification
4. Create Design Document . This may include designs like,
    a. System Architecture Design
    b. Application Architecture Design
    c. GUI Design
    d. API Design
    e. Database Design
    f. Technology Stack
5. Deployment, Test Run & Get Results
6. Prepare Project Report

HONOURS

CS1H40A	CYBER FORENSICS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		VAC	3	1	0	4	2020

COURSE OVERVIEW

The course on Cyber Forensics aims at exploring the basics of Cyber Forensics and Cyber security, the forensic investigation process and principles and the different types of cybercrimes and threats. This course also focuses on the forensic analysis of File systems, the Network, the Windows and Linux Operating systems. The course gives a basic understanding of the forensics analysis tools and a deep understanding of Anti forensics practices and methods. All the above aspects are dealt with case studies of the respective areas.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Explain thebasic concepts in Cyber Forensics, Forensics Investigation Process and Cyber security	Understand
CO2	Infer the basic concepts of File Systems and its associated attribute definitions	Understand
CO3	Utilize the methodologies used in data analysis and memory analysis for detection of artefacts.	Apply
CO4	Identify web attacks and detect artefacts using OWASP and penetration testing.	Apply
CO5	Summarize anti-forensics practices and data hiding methods .	Understand

SYLLABUS

Computer Forensics: History of computer forensics, preparing for computer investigations, understanding Public and private investigations- Forensics Investigation Principles - Forensic Protocol for Evidence Acquisition - Digital Forensics -Standards and Guidelines - Digital Evidence –Data Acquisition - storage formats for digital evidence, determining the best acquisition method, contingency planning for image acquisitions, Cyber Forensics tools- Challenges in Cyber Forensics, Skills Required to Become a Cyber Forensic Expert

Cyber Security: Cybercrimes, Types of Cybercrimes - Recent Data Breaches - Recent Cyber security Trends - Case Study: Sim Swapping Fraud, ATM Card Cloning, Hacking email for money, Google Nest Guard, Email Crimes, Phishing, Types of Phishing.

File system Analysis: FAT and NTFS concepts and analysis -File system category, Content

category, Metadata category, File name category, Application category,Application-level search techniques, Specific file systems, File recovery, Consistency check. FAT data structure-Boot sector, FAT 32 FS info, directory entries, Long file name directory entries.

Windows Forensics: Live Response- Data Collection- Locard's Exchange Principle, Order of Volatility Volatile and Non Volatile Data Live-Response Methodologies: Data Analysis- Agile Analysis, Windows Memory Analysis, Rootkits and Rootkit detection.

Linux Forensics: Live Response Data Collection- Prepare the Target Media, Format the Drive, Gather Volatile Information, Acquiring the Image, Initial Triage, Data Analysis- Log Analysis, Keyword Searches, User Activity, Network Connections, Running Processes, Open File Handlers, The Hacking Top Ten, Reconnaissance Tools

Network Forensics: The OSI Model, Forensic Footprints, Seizure of Networking Devices, Network Forensic Artifacts, ICMP Attacks, Drive-By Downloads, Network Forensic Analysis Tools, Case Study: Wireshark. Web Attack Forensics: OWASP Top 10, Web Attack Tests, Penetration Testing.

Anti-Forensics: Anti-forensic Practices - Data Wiping and Shredding- Data Remanence, Degaussing, Case Study: USB Oblivion, Eraser - Trail Obfuscation: Spoofing, Data Modification, Case Study: Timestamp – Encryption, Case Study: VeraCrypt, Data Hiding: Steganography and Cryptography, Case Study: SilentEye, Anti-forensics Detection Techniques, Case Study: Stegdetect.

TEXT BOOKS

1. Bill Nelson, Amelia Phillips and Christopher Steuart, Computer forensics - Guide to Computer Forensics and Investigations, 4/e, Course Technology Inc.
2. Brian Carrier, File System Forensic Analysis, Addison Wesley, 2005.
3. Harlan Carvey, Windows Forensic Analysis DVD Toolkit, 2/e, Syngress.
4. Cory Altheide, Todd Haverkos, Chris Pogue,Unix and Linux Forensic Analysis DVD Toolkit, 1/e, Syngress.
5. William Stallings,Network Security Essentials Applications and Standards, 4/e, Prentice Hall
6. Eric Maiwald, Fundamentals of Network Security, McGraw-Hill, 2004.

REFERENCES

1. Michael. E. Whitman, Herbert. J. Mattord, Principles of Information Security, Course Technology, 2011.
2. William Stallings, Cryptography and Network Security Principles and Practice, 4/e, Prentice Hall.
3. Niranjan Reddy, Practical Cyber Forensics: An Incident-Based Approach to Forensic Investigations, Apress, 2019.

COURSE PLAN

Module	Contents	No. of Hours
1	**Cyber Forensics and Cyber Security** History of computer forensics, preparing for computer investigations, Understanding Public and private investigations- Forensics Investigation Principles, Forensic Protocol for Evidence Acquisition,	12

	Digital Forensics -Standards and Guidelines - Digital Evidence, Data Acquisition - storage formats for digital evidence, determining the best acquisition method, Contingency planning for image acquisitions, Cyber Forensics tools, Challenges in Cyber Forensics, Skills Required to Become a Cyber Forensic Expert, Cybercrimes, Types of Cybercrimes - Recent Data Breaches - Recent Cyber security Trends, Case Study: Sim Swapping Fraud, ATM Card Cloning, Case Study:Hacking email for money, Google Nest Guard, Email Crimes, Phishing, Types of Phishing	
2	**File System Forensics** FAT and NTFS concepts and analysis, File system category, Content category, Metadata category, File name category,Application category, Application-level search techniques, Specific file systems, File recovery, Consistency check, FAT data structure-Boot sector, FAT 32 FS info, directory entries, Long file name directory entries	12
3	**Operating System Forensics** Live Response- Data Collection- Locard's Exchange Principle, Order of Volatility, Volatile and Non Volatile Data, Live-Response Methodologies: Data Analysis- Agile Analysis, Windows Memory Analysis, Rootkits and Rootkit detection, Linux Forensics: Live Response Data Collection, Prepare the Target Media, Format the Drive, Gather Volatile Information, Acquiring the Image, Initial Triage, Data Analysis- Log Analysis, Keyword Searches, User Activity, Data Analysis- Network Connections, Running Processes, Open File Handlers, The Hacking Top Ten, Reconnaissance Tools	12
4	**Network Forensics** OSI Model, Forensic Footprints, Seizure of Networking Devices, Network Forensic Artifacts, ICMP Attacks, Drive-By Downloads, Network Forensic Analysis Tools, Web Attack Forensics, OWASP Top 10, Web Attack Tests, Penetration Testing-1, Penetration Testing.-2	12
5	**Anti-Forensics** Anti-forensic Practices - Data Wiping and Shredding, Data Remanence, Degaussing, Trail Obfuscation: Spoofing, Data Modification, Role of Encryption in Forensics, Data Hiding: Steganography and Cryptography, Anti-forensics Detection Techniques	12
	Total hours	60

CS1H40B	Reinforcement Learning	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		VAC	3	1	0	4	2020

COURSE OVERVIEW

This course covers fundamental principles and techniques in reinforcement learning. Reinforcement learning is concerned with building programs that learn how to predict and act in a stochastic environment, based on past experience. Applications of reinforcement learning range from classical control problems, such as power plant optimization or dynamical system control, to game playing, inventory control, and many other fields. Topics include Markov decision process, dynamic programming, Monte Carlo, temporal difference, function approximation reinforcement learning algorithms, and applications of reinforcement learning. This course enables the learners to apply reinforcement learning on real world applications and research problems.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO 1	Solve computational problems using probability and random variables.	Apply
CO 2	Apply policy iteration and value iteration reinforcement learning algorithms.	Apply
CO 3	Employ Monte Carlo reinforcement learning algorithms.	Apply
CO 4	Apply temporal-difference reinforcement learning algorithms.	Apply
CO 5	Apply on-policy and off-policy reinforcement learning algorithms with function approximation.	Apply

SYLLABUS

Review Of Probability Concepts: Probability concepts review - Axioms of probability, concepts of random variables, probability mass function, probability density function, cumulative density functions, Expectation. Concepts of joint and multiple random variables, joint, conditional and marginal distributions. Correlation and independence.

Markov Decision Process: Introduction to Reinforcement Learning(RL) terminology - Examples of RL, Elements of RL, Limitations and Scope of RL.

Finite Markov Decision Processes - The Agent–Environment Interface, Goals and Rewards, Returns and Episodes, Policies and Value Functions, Optimal Policies and Optimal Value Functions.

Prediction And Control: Dynamic Programming - Policy Evaluation (Prediction), Policy Improvement, Policy Iteration, Value Iteration.Monte Carlo (MC) for model free prediction and control - Monte Carlo Prediction, MonteCarlo Estimation of Action Values, Monte Carlo Control, Monte Carlo Control without Exploring Starts, Off-policy Prediction via Importance Sampling, Incremental Implementation, Off-policy Monte Carlo Control.

Temporal-Difference (TD) Methods For Model Free Prediction And Control: TD Methods - TD Prediction, Advantages of TD Prediction Methods, Optimality of TD(0), Sarsa: On-policy TD Control, Q-learning: Off-policy TD Control, Expected Sarsa.n-step Bootstrapping- n-step TD Prediction, n-step Sarsa, step Off-policy Learning, Off-policy Learning Without Importance Sampling: The n-step Tree Backup Algorithm.

Function Approximation Method: On-policy Prediction with Approximation - Value-function Approximation, The Prediction Objective, Stochastic-gradient Methods, Linear Methods.Eligibility Traces - The **λ**-return, TD(**λ**), n-step Truncated **λ**-return Methods, Sarsa(**λ**).Policy Gradient Methods - Policy Approximation and its Advantages, The Policy Gradient Theorem, REINFORCE: Monte Carlo Policy Gradient, REINFORCE with Baseline, Actor–Critic Methods.

TEXT BOOKS

1 Richard S. Sutton and Andrew G. Barto, Reinforcement Learning: An Introduction, , 2nd Edition

2 Alberto Leon-Garcia, Probability, Statistics, and Random Processes for Electrical Engineering, 3rd Edition,

REFERENCES

1 Reinforcement Learning: State-of-the-Art, Marco Wiering and Martijn van Otterlo, Eds

2 Algorithms for Reinforcement Learning, Szepesvari (2010), Morgan & Claypool.

3 Artificial Intelligence: A Modern Approach, Stuart J. Russell and Peter Norvig

4 Mathematical Statistics and Data Analysis by John A. Rice,University of California, Berkeley, Third edition, published by Cengage.

5 Machine Learning: A Probabilistic Perspective, Kevin P. Murphy

COURSE PLAN

Module	Contents	No. of Hours
1	**Review of Probability Concepts** Axioms of probability, concepts of random variables, Probability mass function, Probability density function, Cumulative density functions, Expectation of random variables, Joint and multiple random variables, Conditional and marginal distributions Correlation and independence	12
2	**Markov Decision Process** Introduction to Reinforcement Learning(RL) terminology - Examples of RL, Elements of RL, Limitations and Scope of RL, Finite Markov Decision Processes, The Agent Environment Interface Goals and Rewards, Returns and Episodes, Policies and Value Functions, Optimal Policies and Optimal Value Functions, Optimal Policies and Optimal Value Functions	12
3	**Prediction And Control** Policy Evaluation (Prediction), Policy Improvement, Policy Iteration, Value Iteration, Monte Carlo Prediction, Monte Carlo Estimation of Action Values, Monte Carlo Control, Monte Carlo Control without Exploring Starts, Off-policy Prediction via Importance Sampling, Incremental Implementation, Off-policy Monte Carlo Control	12
4	**Temporal-Difference (Td) Methods TB-1 (Ch 6,7)** TD Prediction, Advantages of TD Prediction Methods Optimality of TD (0), Sarsa: On-policy TD Control, Q-learning: Off-policy TD Control, Expected Sarsa, n-step TD Prediction, n-step Sarsa, n-step Off-policy Learning Off-policy Learning Without Importance Sampling: The n-step Tree Backup Algorithm	12
5	**Function Approximation Method TB-1 (Ch 9,12,13)** Value-function Approximation, The Prediction Objective, Stochastic-gradient Methods, Linear Methods, The Lambda-return , TD(Lambda), n-step Truncated Lambda-return Methods, Sarsa(Lambda), Policy Approximation and its Advantages, The Policy Gradient Theorem, REINFORCE: Monte Carlo Policy Gradient REINFORCE with Baseline, Actor–Critic Methods	12
	Total hours	**60**

CS1H40 C	LOGIC FOR COMPUTER SCIENCE	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		VAC	3	1	0	4	2020

COURSE OVERVIEW

This course enables the learners to understand the concepts of various logics used in computer science. The course covers the standard and most popular logics such as propositional logic, predicate logic, linear temporal logic, computation tree logic, Hoare logic and modal logic. This course helps the students to develop solutions for specification and verification of real world systems.

COURSE OUTCOMES

After the completion of the course the student will be able to

Course Outcomes	Description	Level
CO1	Explain the concepts of Predicate Logic, Propositional Logic, Linear Temporal Logic, Computation Tree Logic, Hoare Logic and Modal Logic as a formal language.	Understand
CO2	Develop proofs to show the satisfiability, validity and equivalence of logic formulas.	Apply
CO3	Illustrate model checking and program verification to prove correctness of systems	Apply
CO4	Demonstrate *Alloy Analyzer* to model and analyze software systems.	Apply
CO5	Demonstrate *New Symbolic Model Verifier (NuSMV)* as a model checking tool to check the validity of temporal logic formulas.	Apply

SYLLABUS

Propositional Logic: Declarative Sentences, Natural Deduction, Propositional Logic as a Formal Language, Semantics of Propositional Logic, Normal Forms, SAT Solvers.

Predicate Logic: The Need of a Richer Language, Predicate Logic as a Formal Language, Proof Theory of Predicate Logic, Semantics of Predicate Logic, Undesirability of Predicate Logic, Expressiveness of Predicate Logic.

Verification by Model Checking: Motivation for Verification, Linear Time Temporal Logic (LTL), Model Checking Systems, Tools, Properties, Branching Time Logic, Computation Tree Logic (CTL) and the Expressive Powers of LTL and CTL, Model Checking Algorithms, The Fixed Point Characterization of CTL.

Program Verification: Why Should We Specify and Verify Code, A Framework for Software Verification, Proof Calculus for Partial Correctness, Proof Calculus for Total Correctness, Programming by Contract.

Modal Logics and Agents: Modes of Truth, Basic Modal Logic, Logic Engineering, Natural Deduction, Reasoning about Knowledge in a Multi-Agent System.

TEXT BOOKS

1. Michael Huth and Mark Ryan, Logic in Computer Science, 2/e, Cambridge University Press, 2004.

REFERENCES

1. Daniel Jackson, Software Abstractions, MIT Press, 2011.
2. Roberto Cavada, Alessandro Cimatti, Gavin Keighren, Emanuele Olivetti, Marco Pistore and Marco Roveri, NuSMV 2.6 Tutorial (available at *https://nusmv.fbk.eu*).
3. Tutorial for Alloy Analyzer 4.0 (available at *https://alloytools.org/tutorials/online/*).

COURSE PLAN

Module	Contents	No.of Hours
1	**Propositional Logic** Declarative Sentences, Natural Deduction, Rule for Natural Deduction, Derived Rules, Natural Deduction in Summary Provable Equivalence, Proof by Contradiction. Propositional Logic as a Formal language, Semantics of Propositional Logic – The Meaning of Logical Connectives, Soundness of Propositional Logic, Completeness of Propositional Logic (Proof not required), Semantic Equivalence, Satisfiability and Validity, Normal Forms – Conjunctive Normal Forms and Validity, Horn Clauses and Satisfiability, SAT Solvers – A Linear Solver, A Cubic Solver	12
2	**Predicate Logic** The Need of a Richer language, Predicate Logic as a Formal Language –Terms, Formulas, Free and Bound Variables, Substitution, Proof Theory of Predicate Logic – Natural Deduction Rules, Proof Theory of Predicate Logic – Quantifier Equivalences, Semantics of Predicate Logic – Models, Semantic Entailment, The Semantics of Equality, Undecidabilty of Predicate Logic (no proof required), Expressiveness of Predicate Logic – Existential Second Order Logic, Universal Second Order Logic, Micromodels of Software – State Machines, A Software Micromodel (Alloy) (Lecture 1), A Software Micromodel (Alloy) (Lecture 2)	12
3	**Verification by Model Checking** Motivation for Verification, Linear Time Temporal Logic (LTL) – Syntax, Semantics of LTL – Practical Patterns of Specifications, Important Equivalences between LTL Formulas, Adequate Sets of Connectives for LTL,Introduction to model checking, Model Checking Systems, Tools, Properties, Model checking example: Mutual Exclusion The NewSymbolic Model Verifier (NuSMV) Model Checker –Introduction, Mutual Exclusion Revisited, The NuSMV Model Checker – The Ferryman, The Alternating Bit Protocol, Branching Time Logic – Syntax of Computation Tree Logic (CTL), Semantics of CTL, Practical Patterns of Specification, Important Equivalences between	12

	CTL Formulas, Adequate Sets of CTL Connectives, CTL and the Expressive Powers of LTL and CTL – Boolean Combinations of Temporal Formulas in CTL, Model-Checking Algorithms – The CTL Model Checking Algorithm CTL Model Checking with Fairness, The LTL Model Checking Algorithm (Algorithm only)	
4	**Program Verification** Introduction to Program Verification, Need of Specification and Verification of Code, A Framework for Software Verification – A Core Programming Language, Hoare Triples, A Framework for Software Verification – Partial and Total Correctness, Program Variables and Logical Variables, Proof Calculus for partial Correctness – Proof Rules, Proof Calculus for partial Correctness – Proof Tableaux, Proof Calculus for partial Correctness – A Case Study: Minimal-Sum Section, Proof Calculus for Total Correctness, Programming by Contract	12
5	**Modal Logics and Agents** Modes of Truth, basic Modal Logic – Syntax, Basic Modal Logic – Semantics, Logic Engineering – The Stock of Valid Formulas, Important Properties of the Accessibility Relation, Logic Engineering – Correspondence Theory, Some Modal Logics, Natural Deduction Reasoning about Knowledge in a Multi-Agent System –Examples (The Wise - Man Puzzle, The Muddy – Children Puzzle), The Modal Logic KT45n, Natural Deduction for KT45n Formalizing the Examples (The Wise - Man Puzzle, The Muddy – Children Puzzle)	12
	Total hours	**60**

SEMESTER VIII

CS1U40B	DISTRIBUTED COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	2	1	0	3	2020

COURSE OVERVIEW:

The purpose of this course is to understand the system models, algorithms and protocols that allow computers to communicate and coordinate their actions to solve a problem. This course helps the learner to understand the distributed computation model and various concepts like global state, termination detection, mutual exclusion, deadlock detection, shared memory, failure recovery, consensus, file system. It helps the learners to develop solutions to problems in distributed computing environment.

COURSE OUTCOMES

After the completion of the course the student will be able to

CO1	Summarize various aspects of distributed computation model and logical time.	Understand
CO2	Illustrate election algorithm, global snapshot algorithm and termination detection algorithm.	Apply
CO3	Compare token based, non-token based and quorum based mutual exclusion algorithms.	Understand
CO4	Recognize the significance of deadlock detection and shared memory in distributed systems	Understand
CO5	Explain the concepts of failure recovery and consensus.	Understand
CO6	Illustrate distributed files system architectures.	Understand

SYLLABUS

Module – 1 (Distributed systems basics and Computation model)

Distributed System **–** Definition, Relation to computer system components, Motivation, Primitives for distributed communication, Design issues, Challenges and applications. A model of distributed computations **–** Distributed program, Model of distributed executions, Models of communication networks, Global state of a distributed system, Cuts of a distributed computation, Past and future cones of an event, Models of process communications.

Module – 2 (Election algorithm, Global state and Termination detection)

Logical time – A framework for a system of logical clocks, Scalar time, Vector time. Leader election algorithm – Bully algorithm, Ring algorithm. Global state and snapshot recording algorithms –System model and definitions, Snapshot algorithm for FIFO channels – Chandy Lamport algorithm. Termination detection – System model of a distributed computation, Termination detection using distributed snapshots, Termination detection by weight throwing,Spanning-tree-based algorithm.

Module – 3 (Mutual exclusion and Deadlock detection)

Distributed mutual exclusion algorithms – System model, Requirements of mutual exclusion algorithm. Lamport's algorithm, Ricart–Agrawala algorithm, Quorum-based mutual exclusion algorithms – Maekawa's algorithm. Token-based algorithm – Suzuki–Kasami's broadcast algorithm. Deadlock detection in distributed systems – System model, Deadlock handling strategies, Issues in deadlock detection, Models of deadlocks.

Module – 4 (Distributed shared memory and Failure recovery)

Distributed shared memory – Abstraction and advantages. Shared memory mutual exclusion – Lamport's bakery algorithm. Check pointing and rollback recovery – System model, consistent and inconsistent states, different types of messages, Issues in failure recovery, checkpoint based recovery, log based roll back recovery.

Module – 5 (Consensus and Distributed file system)

Consensus and agreement algorithms – Assumptions, The Byzantine agreement and other problems, Agreement in (message-passing) synchronous systems with failures – Consensus algorithm for crash failures. Distributed file system – File service architecture, Case studies: Sun Network File System, Andrew File System, Google File System.

TEXT BOOKS

1. Ajay D. Kshemkalyani and Mukesh Singhal, Distributed Computing: Principles, Algorithms, and Systems, Cambridge University Press, 2011.

REFERENCES

1. George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair. Distributed Systems: Concepts and Design, Addison Wesley, Fifth edition.
2. Kai Hwang, Geoffrey C Fox, Jack J Dongarra, Distributed and Cloud Computing – From Parallel Processing to the Internet of Things, Morgan Kaufmann Publishers, 2012.
3. Sukumar Ghosh, Distributed Systems: An Algorithmic Approach, CRC Press, Second edition, 2015.
4. Maarten Van Steen, Andrew S. Tanenbaum, Distributed Systems, Prentice Hall of India,Third edition, 2017.
5. Randy Chow and Theodore Johnson, Distributed Operating Systems and Algorithm Analysis, Pearson Education India, First edition, 2009.
6. Valmir C. Barbosa, An Introduction to Distributed Algorithms, MIT Press, 2003.

COURSE PLAN

Module	Contents	No. of Hours
1	**Distributed systems basics and Computation model** Distributed System – Definition, Relation to computer system components , Primitives for distributed communication, Design issues, challenges and applications, Design issues, challenges and applications, A model of distributed computations – Distributed program, Model of distributed executions, Models of communication networks, Global state of a distributed system, Cuts of a distributed computation, Cuts of a distributed computation, Past and future cones of an event, Models of process communications.	**9**
2	**Election algorithm, Global state and Termination detection** Logical time – A framework for a system of logical clocks, Scalar time, Vector time, Leader election algorithm – Bully Algorithm, Ring Algorithm, Global state and snapshot recording algorithms – System model and definitions, Snapshot algorithm for FIFO channels – Chandy Lamport algorithm, Termination detection – System model of a distributed computation, Termination detection using distributed snapshots, Termination detection by weight throwing, Spanning tree-based algorithm	**9**
3	**Mutual exclusion and Deadlock detection** Distributed mutual exclusion algorithms – System model, Lamport's algorithm, Ricart–Agrawala algorithm, Quorum-based mutual exclusion algorithms – Maekawa's algorithm, Token-based algorithm – Suzuki–Kasami's broadcast algorithm, Deadlock detection in distributed systems – System model, Deadlock handling strategies, Issues in deadlock detection Models of deadlocks	**9**
4	**Distributed shared memory and Failure recovery** Distributed shared memory – Abstraction and advantages, shared memory mutual exclusion – Lamport's bakery algorithm, Checkpointing and rollback recovery – System model, consistent and inconsistent states different types of messages, Issues in failure recovery, checkpoint based recovery, log based roll back recovery,log based roll back recovery	**9**
5	**Consensus and Distributed file system** Consensus and agreement algorithms – Assumptions, The Byzantine agreement and other problems, agreement in (message-passing) synchronous systems with failures –Consensus algorithm for crash failures, Agreement in (message-passing) synchronous systems with failures –Consensus algorithm for crash failures,Distributed File System – File Service Architecture, Case Studies: Sun Network File System, Andrew File System.	**9**
		45

CS1U40C	COMPREHENSIVE COURSE VIVA	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PCC	1	0	0	1	2020

COURSE OVERVIEW:

The objective of this Course viva is to ensure the basic knowledge of each student in the most fundamental core courses in the curriculum. The viva voce shall be conducted based on the core subjects studied from third to eighth semester. This course helps the learner to become competent in placement tests and other competitive examinations.

Guidelines

1. The course should be mapped with a faculty and classes shall be arranged for practicing questions based on the core courses listed in the curriculum.
2. The viva voce will be conducted by the same three member committee assigned for final project phase II evaluation. It comprises of Project coordinator, expert from Industry/research Institute and a senior faculty from a sister department.
3. The pass minimum for this course is 25.
4. The mark will be treated as internal and should be uploaded along with internal marks of other courses.
5. Comprehensive Viva should be conducted along with final project evaluation by the three member committee.

Mark Distribution

Total marks: 50, only CIE, minimum required to pass : **25**

CS1U49C	PROJECT PHASE II	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PWS	0	0	12	4	2020

COURSE OVERVIEW:

The course 'Project Work' is mainly intended to evoke the innovation and invention skills in a student. The course will provide an opportunity to synthesize and apply the knowledge and analytical skills learned, to be developed as a prototype or simulation. The project extends to 2 semesters and will be evaluated in the 7th and 8th semester separately, based on the achieved objectives. One third of the project credits shall be completed in 7th semester and two third in 8th semester. It is recommended that the projects may be finalized in the thrust areas of the respective engineering stream or as interdisciplinary projects. Importance should be given to address societal problems and developing indigenous technologies.

COURSE OUTCOMES: After successful completion of the course, the students will be able to:

CO1	Model and solve real world problems by applying knowledge across domains	Apply
CO2	Develop products, processes or technologies for sustainable and socially relevant applications	Apply
CO3	Function effectively as an individual and as a leader in diverse teams and to comprehend and execute designated tasks	Apply
CO4	Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms	Apply
CO5	Identify technology/research gaps and propos innovative/creative solutions	Analyze
CO6	Organize and communicate technical and scientific findings effectively in written and oral forms	Apply

Phase 2

TARGETS:

- In depth study of the topic assigned in the light of the report prepared under Phase - I;
- Review and finalization of the approach to the problem relating to the assigned topic.
- Preparing a detailed action plan for conducting the investigation, including teamwork.
- Detailed Analysis/ Modeling / Simulation/ Design/ Problem Solving/Experiment as needed.
- Final development of product/ process, testing, results, conclusions and future directions.
- Preparing a paper for Conference Presentation/ Publication in Journals, if possible.
- Presenting projects in Project Expos conducted by the University at the cluster level and/ or state level as well as others conducted in India and abroad.
- Filing Intellectual Property Rights (IPR) if applicable.
- Preparing a report in the standard format for being evaluated by the Department

Assessment Board.

- Final project presentation and viva voce by the assessment board including the external expert.

Evaluation Guidelines & Rubrics

Total: 150 marks (Minimum required to pass: 75 marks).

- Project progress evaluation by guide: 30 Marks.
- Two interim evaluations by the Evaluation Committee: 50 Marks (25 marks for each evaluation).
- Final evaluation by the Final Evaluation committee: 40 Marks
- Quality of the report evaluated by the evaluation committee: 30 Marks

(The evaluation committee comprises HoD or a senior faculty member, Project coordinator and project supervisor. The final evaluation committee comprises of Project coordinator, expert from Industry/research/academic Institute and a senior faculty from a sister department).

CS1U42A	**DEEP LEARNING**	**CATEGORY**	**L**	**T**	**P**	**CREDIT**	**YEAR OF INTRODUCTION**
		PEC	**2**	**1**	**0**	**3**	**2020**

COURSE OVERVIEW:

Deep Learning is the recently emerged branch of machine learning, particularly designed to solve a wide range of problems in Computer Vision and Natural Language Processing. In this course, the building blocks used in deep learning are introduced. Specifically, neural networks, deep neural networks, convolutional neural networks and recurrent neural networks. Learning and optimization strategies such as Gradient Descent, Nesterov Accelerated Gradient Descent, Adam, AdaGrad and RMSProp are also discussed in this course. This course will helps the students to attain sound knowledge of deep architectures used for solving various Vision and NLP tasks. In future, learners can master modern techniques in deep learning such as attention mechanisms, generative models and reinforcement learning.

COURSE OUTCOMES

After the completion of the course, the student will be able to

CO1	Illustrate the basic concepts of neural networks and its practical issues	Apply
CO2	Outline the standard regularization and optimization techniques for deep neural network	Understand
CO3	Implement the foundation layers of CNN (pooling, convolutions	Apply
CO4	Implement a sequence model using recurrent neural networks	Apply
CO5	Use different neural network/deep learning models for practical applications.	Apply

SYLLABUS

Module-1 (Neural Networks)

Introduction to neural networks -Single layer perceptrons, Multi-Layer Perceptrons (MLPs), Representation Power of MLPs, Activation functions - Sigmoid, Tanh, ReLU, Softmax. , Risk minimization, Loss function, Training MLPs with backpropagation, Practical issues in neural network training - The Problem of Overfitting, Vanishing and exploding gradient problems, Difficulties in convergence, Local and spurious Optima, Computational Challenges. Applications of neural networks.

Module-2 (Deep learning)

Introduction to deep learning, Deep feed forward network, Training deep models, Optimization techniques - Gradient Descent (GD), GD with momentum, Nesterov accelerated GD, Stochastic GD, AdaGrad, RMSProp, Adam. Regularization Techniques - L1 and L2 regularization, Early stopping, Dataset augmentation, Parameter sharing and tying, Injecting noise at input, Ensemble methods, Dropout, Parameter initialization.

Module-3 (Convolutional Neural Network)

Convolutional Neural Networks – convolution operation, motivation, pooling, Convolution and Pooling as an infinitely strong prior, variants of convolution functions, structured outputs, data types, efficient convolution algorithms.

Module- 4 (Recurrent Neural Network)

Recurrent neural networks – Computational graphs, RNN design, encoder – decoder sequence to sequence architectures, deep recurrent networks, recursive neural networks, modern RNNs LSTM and GRU.

Module-5 (Application Areas)

Applications – computer vision, speech recognition, natural language processing, common word embedding: continuous Bag-of-Words, Word2Vec, global vectors for word representation (GloVe). Research Areas – autoencoders, representation learning, boltzmann machines, deep belief networks.

TEXT BOOKS

1. Goodfellow, I., Bengio,Y., and Courville, A., Deep Learning, MIT Press, 2016.
2. Neural Networks and Deep Learning, Aggarwal, Charu C.
3. Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms (1st. ed.). Nikhil Buduma and Nicholas Locascio. 2017. O'Reilly Media, Inc

REFERENCES

1. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.
2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
3. Michael Nielsen, Neural Networks and Deep Learning, 2018

CS1U42B	PROGRAMMING PARADIGMS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

The course provides the learners a clear understanding of the main constructs of contemporary programming languages and the various systems of ideas that have been used to guide the design of programming languages. This course covers the concepts of Names, Bindings & Scope, Statement-Level Control Structures, Sub Programs, Support for Object Oriented Programming, Exception Handling, Concurrency Control, Functional Programming and Logic Programming. This course helps the learners to equip with the knowledge necessary for the critical evaluation of existing and upcoming programming languages. It also enables the learner to choose the most appropriate language for a given programming task, apply that language's approach to structure or organize the code, classify programming languages based on their features and to design new generation languages.

COURSE OUTCOMES

CO1	Explain the criteria for evaluating programming languages and compare Imperative, Functional and Logic programming languages	Understand
CO2	Illustrate the characteristics of data types and variables	Apply
CO3	Comprehend how control flow structures and subprograms help in developing the structure of a program to solve a computational problem	Apply
CO4	Explain the characteristics of Object-Oriented Programming Languages	Understand
CO5	Compare concurrency constructs in different programming languages	Understand

SYLLABUS

Module – 1

Introduction – Role of Programming Languages, Programming Domains, Language Evaluation Criteria, Influence on Language Design, Language Design Trade-offs, Implementation Methods. Names, Bindings & Scope – Names, Variables, Concept of Binding, Scope and Lifetime, Referencing Environments.

Module - 2

Data Types – Primitive Data Types, Character String Types, User-Defined Ordinal Types, Array Types, Record Types, List Types, Pointer & Reference Types, Type Checking, Strong Typing, Type Equivalence. Expressions – Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation. Assignment -Assignment Statements, Mixed-mode Assignment.

Module - 3

Statement-Level Control Structures – Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands. Subprograms – Design Issues of Subprograms, Local Referencing Environments, Parameter Passing Methods, Subprograms as Parameters, Overloaded Subprograms, Closures, Co-routines

Module - 4

Support for Object Oriented Programming – Inheritance, Dynamic Binding, Design Issues for Object Oriented Languages, Support for Object Oriented Programming in C++, Implementation of Object-oriented Constructs. Exception Handling – Basic Concepts, Design Issues.

Module - 5

Concurrency – Subprogram Level Concurrency, Semaphores, Monitors, Message Passing. Functional Programming Languages – Introduction to LISP and Scheme, Comparison of Functional and Imperative Languages. Logic Programming Languages – Basic Elements of Prolog, Applications of Logic Programming.

TEXT BOOKS

1. Robert W Sebesta, Concepts of Programming Languages, 10th Edition, Pearson.
2. Scott M L, Programming Language Pragmatics, 3rd Edition, Morgan Kauffman Publishers.

REFERENCES

1. Kenneth C. Louden, Programming Languages: Principles and Practice, 2nd Edition, Cengage Learning.
2. Tucker A. B. and R. E. Noonan, Programming Languages: Principles and Paradigms, 2nd Edition. –TMH.
3. Ravi Sethi, Programming Languages: Concepts & Constructs, 2nd Edition., Pearson Education.
4. David A. Watt, Programming Language Design Concepts, Wiley Dreamtech.

COURSE PLAN

Module	Contents	No. of Hours
1	Introduction: Reasons for studying Concepts of programming languages, Programming Domains, Language Evaluation Criteria, Influence on Language Design, Language Design Trade-offs, Implementation Methods ,Names, Variables, Concept of Binding, Scope and Lifetime, Referencing Environments	**9**
2	Primitive Data Types, Character String Types, User-Defined Ordinal Types, Array Types, Record Types, List Types, Pointer and Reference Types, Implementation of pointer and reference types, Type Checking, Strong Typing, Type Equivalence, Expressions and Assignment Statements, Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation, Assignment Statements, Mixed-mode Assignment	**9**
3	Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands, Subprograms: Design Issues of Subprograms, Local Referencing Environments, Parameter Passing Methods, Subprograms as Parameters, Overloaded Subprograms, Closures, Co-routines	**9**
4	Inheritance, Dynamic Binding, Design Issues for Object Oriented Languages, Support for Object Oriented Programming in C++, Implementation of Object-Oriented Constructs, Exception Handling – Basic Concepts, Exception Handling - Design Issues	**9**
5	Subprogram Level Concurrency, Semaphores, Monitors, Message Passing, Introduction to LISP and Scheme, Comparison of Functional and Imperative Languages, Basic Elements of Prolog, Applications of Logic Programming	**9**
		45

CS1U42C	CRYPTOGRAPHY	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	**2**	**1**	**0**	**3**	**2020**

COURSE OVERVIEW:

This course helps the learners to explore the fundamental concepts of symmetric and asymmetric cipher models. This course covers fundamental concepts of authentication protocols, network security protocols and web security protocols. The concepts covered in this course enable the learners in effective use of cryptographic techniques for securing network applications.

COURSE OUTCOMES

After the completion of the course, the student will be able to

CO1	Summarize different classical encryption techniques	Understand
CO2	Identify mathematical concepts for different cryptographic algorithms	Understand
CO3	Demonstrate cryptographic algorithms for encryption/key exchange.	Apply
CO4	Summarize different authentication and digital signature schemes.	Understand
CO5	Identify security issues in network, transport and application layers and outline appropriate security protocols.	Understand

SYLLABUS

Module-1 (Authentication Protocols)

Symmetric Cipher Models- Substitution techniques- Transposition techniques- Rotor machines-Steganography. Simplified DES- Block Cipher principles- The Data Encryption Standard, Strength of DES- Differential and linear Cryptanalysis. Block Cipher Design principles- Block Cipher modes of operations.

Module-2 (E- mail Security)

IDEA: Primitive operations- Key expansions- One round, Odd round, Even Round- Inverse keys for decryption. AES: Basic Structure- Primitive operation- Inverse Cipher- Key Expansion,
Rounds, Inverse Rounds. Stream Cipher –RC4.

Module-3 (Network Layer Security and Web Security)

Public key Cryptography: - Principles of Public key Cryptography Systems, Number theory- Fundamental Theorem of arithmetic, Fermat's Theorem, Euler's Theorem, Euler's Totient Function, Extended Euclid's Algorithm, Modular arithmetic. RSA algorithm- Key Management - Diffie-Hellman Key Exchange, Elliptic curve cryptography.

Module-4 (Real-time Security and Application Layer Security)

Authentication requirements- Authentication functions- Message authentication codes- Hash

functions- SHA -1, MD5, Security of Hash functions and MACs- Authentication protocols- Digital signatures-Digital signature standards.

Module-5 (System Security and Wireless Security)

Network security: Electronic Mail Security: Pretty good privacy- S/MIME. IP Security: Architecture - authentication Header-Encapsulating Security payload- Combining Security associations- Key management. Web Security: Web Security considerations- secure Socket Layer and Transport layer Security- Secure electronic transaction. Firewalls-Packet filters- Application Level Gateway- Encrypted tunnels

TEXT BOOKS

1. Behrouz A. Forouzan, Cryptography and Network Security, Tata McGraw-Hill. 2010
2. William Stallings, Cryptography and Network Security, Pearson Education, 2014.

REFERENCES

1. B. Schneier , Applied Cryptography, Protocols, Algorithms, and Source Code in C, 2 nd Edn, Wiley, 1995.
2. Charlie Kaufman, Radia Perlman, Mike Speciner, Network Security, PHI, 2002

COURSE PLAN

Module	Contents	No.of Hours
1	Symmetric Cipher Models- Substitution techniques- Transposition techniques- Rotor machines-Steganography. Simplified DES- Block Cipher principles- The Data Encryption Standard, Strength of DES- Differential and linear Cryptanalysis. Block Cipher Design principles- Block Cipher modes of operations.	**9**
2	IDEA: Primitive operations- Key expansions- One round, Odd round, Even Round- Inverse keys for decryption. AES: Basic Structure- Primitive operation- Inverse Cipher- Key Expansion, Rounds, Inverse Rounds. Stream Cipher –RC4.	**9**
3	Public key Cryptography: - Principles of Public key Cryptography Systems, Number theory- Fundamental Theorem of arithmetic, Fermat's Theorem, Euler's Theorem, Euler's Totient Function, Extended Euclid's Algorithm, Modular arithmetic. RSA algorithm- Key Management - Diffie-Hellman Key Exchange, Elliptic curve cryptography	**9**
4	Authentication requirements- Authentication functions- Message authentication codes- Hash functions- SHA -1, MD5, Security of Hash functions and MACs- Authentication protocols-Digital signatures-Digital signature standards.	**9**
5	Network security: Electronic Mail Security: Pretty good privacy- S/MIME. IP Security: Architecture-authentication Header- Encapsulating Security payload- Combining Security associations- Key management. Web Security: Web Security considerations- secure Socket Layer and Transport layer Security- Secure electronic transaction. Firewalls-Packet filters- Application Level Gateway- Encrypted tunnels.	**9**
		45

CS1U42D	SOFT COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

This course enables the learners to understand the concepts of Soft Computing techniques and its applications. It covers Artificial Neural Networks, operations and models of fuzzy logic, genetic algorithms and multi objective optimization techniques. This course helps the students to develop algorithms and solutions for different real world applications.

COURSE OUTCOMES

CO1	Describe soft computing techniques and the basic models of Artificial Neural Network	Understand
CO2	Solve practical problems using neural networks	Apply
CO3	Illustrate the operations, model and applications of fuzzy logic	Apply
CO4	Illustrate the concepts of Genetic Algorithm	Apply
CO5	Describe the concepts of multi-objective optimization models and the need for using hybrid soft computing approaches	Understand

SYLLABUS

Module – 1 (Introduction to Soft Computing & Artificial Neural Network)

Introduction to Soft Computing. Difference between Hard Computing & Soft Computing. Applications of Soft Computing. Artificial Neurons Vs Biological Neurons. Basic models of artificial neural networks – Connections, Learning, Activation Functions. McCulloch and Pitts Neuron. Hebb network.

Module – 2 (Supervised Learning Network)

Perceptron Networks– Learning rule, Training and testing algorithm. Adaptive Linear Neuron– Architecture, Training and testing algorithm. Back propagation Network – Architecture, Training and testing algorithm.

Module - 3 (Fuzzy Logic & Defuzzification)

Fuzzy sets – properties, operations on fuzzy set. Fuzzy membership functions, Methods of membership value assignments – intuition, inference, Rank Ordering. Fuzzy relations– operations on fuzzy relation. Fuzzy Propositions. Fuzzy implications. Defuzzification– Lamda cuts, Defuzzification methods.

Module - 4 (Fuzzy Inference System & Genetic Algorithm)

Fuzzy Inference Systems - Mamdani and Sugeno types. Fuzzy Logic Controller. Concepts of genetic algorithm. Operators in genetic algorithm - coding, selection, cross over, mutation. Stopping condition for genetic algorithm.

Module - 5 (Multi Objective Optimization & Hybrid Systems)

Multi objective optimization problem. Principles of Multi- objective optimization, Dominance and pareto-optimality. Optimality conditions. Neuro-fuzzy hybrid systems. Genetic – neuro hybrid systems.

TEXT BOOKS

1. S.N.Sivanandam and S.N. Deepa, Principles of Soft Computing , 2nd Edition, John Wiley & Sons.

2. Kalyanmoy Deb, Multi-objective Optimization using Evolutionary Algorithms, 1st Edition, John Wiley & Sons.

REFERENCES

1. Timothy J Ross, Fuzzy Logic with Engineering Applications, John Wiley & Sons, 2016.
2. T.S.Rajasekaran, G.A.Vijaylakshmi Pai "Neural Networks, Fuzzy Logic & Genetic Algorithms Synthesis and Applications", Prentice-Hall India.
3. Simon Haykin, "Neural Networks- A Comprehensive Foundation", 2/e, Pearson Education.
4. Zimmermann H. J, "Fuzzy Set Theory & Its Applications", Allied Publishers Ltd.

COURSE PLAN

Module	Contents	No.of Hours
1	**Introduction to Soft Computing & Artificial Neural Network** Introduction to Soft Computing, Difference between Hard Computing & Soft Computing & Applications of Soft Computing, Artificial Neurons Vs Biological Neurons, Basic models of artificial neural networks, Activation Functions McCulloch and Pitts Neuron, Hebb network	**9**
2	**Supervised Learning Network** Perceptron networks – Learning rule, Training and testing algorithm, Perceptron networks – Problems, Adaptive Linear Neuron (Lecture I),Adaptive Linear Neuron (Lecture II), Adaptive Linear Neuron-Problems (Lecture III), Back propagation Network (Lecture I), Back propagation Network (Lecture II)	**9**
3	**Fuzzy Logic & Defuzzification** Introduction to Fuzzy Set, Properties & operations on fuzzy sets Fuzzy membership functions, Fuzzification, Methods of membership value assignments, Fuzzy relations, Operations on Fuzzy Relation, Fuzzy Propositions & Fuzzy Implications Lamda cuts for fuzzy sets, Defuzzification methods(Lecture I), Defuzzification methods(Lecture II)	**9**
4	**Fuzzy Inference System & Genetic Algorithm** Fuzzy Inference Systems - Mamdani type,Fuzzy Inference Systems - Sugeno type, Fuzzy Logic Controller, Introduction to genetic algorithm, operators in genetic algorithm - coding Selection, Cross over, Mutation, stopping condition for genetic algorithm	**9**
5	**Multi-Objective Optimization & Hybrid System** MOOP-Linear &Non linear, Convex & Non Convex,Principles of MOO-Illustrating Pareto Optimal Solutions, Objectives in MOO,Dominance & Pareto-Optimality-Concept of Domination ,Properties of Dominance Relation, Pareto Optimality, Procedure for finding a non dominated set ,Optimality Conditions, Neuro Fuzzy hybrid system-Classification & characteristics, Genetic – neuro hybrid systems	**9**
		45

CS1U42E	FUZZY SET THEORY AND APPLICATIONS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

This course equips the students to understand the concepts of fuzziness and its use in building better solutions to problems. The course covers basic concepts of fuzzy sets, fuzzy relations, fuzzy logic and building of fuzzy approximation-based solutions. It helps students to design and develop fuzzy based solutions to real world applications.

COURSE OUTCOMES:

After the completion of the course, the student will be able to

CO1	Explain fuzzy logic based problem solving	Understand
CO2	Summarize the concepts of crisp sets, crisp relations, crisp logic with fuzzy sets, fuzzy relations and fuzzy logic.	Apply
CO3	Develop fuzzy systems by selecting appropriate membership functions, fuzzification and defuzzification methods	Apply
CO4	Develop solutions using graphical and rule-based methods	Apply
CO5	Make use of fuzzy logic inference to solve real world problems	Apply

SYLLABUS

Module – 1 (Basic Fuzzy Set Theory)

The case for imprecision, Utility and Limitations of Fuzzy Systems, Fuzzy Sets and Membership, Classical Sets – Properties, Operations, Fuzzy Sets – Properties and Operations, Classical Relations – Cartesian Product, Operations and Properties of Crisp Relations, Composition, Fuzzy Relations – Cardinality, Operations, Properties, Fuzzy Cartesian Product and Composition.

Module – 2 (Fuzzy Membership Functions)

Tolerance and Equivalence Relations – Crisp and Fuzzy, Similarity Methods – Cosine, Min-max, Fuzzy Membership Functions – Features, Fuzzification, Defuzzification to Crisp Sets, λ-Cutsfor Fuzzy Relations, Linguistic Hedges.

Module - 3 (Fuzzification and Defuzzification Methods)

Development of Membership Functions –Intuition, Inference, Rank ordering, Inductive reasoning. Defuzzification to Scalars - Max membership principle, Centroid method, Weighted average method, Mean max membership, Center of sums, Center of largest area, First (or last) of maxima.

Module - 4 (Fuzzy Inference)

Classical Logic, Fuzzy Logic, Approximate Reasoning, Fuzzy (Rule-Based) Systems - Multiple conjunctive antecedents, Multiple disjunctive antecedents, Aggregation of fuzzy rules, Graphical Techniques of Inference.

Module - 5 (Fuzzy Applications)

Applications of Fuzzy Systems - Fuzzy Classification, Fuzzy Pattern Recognition, Fuzzy Control Systems, Fuzzy Systems and Neural Networks, Fuzzy Clustering, Fuzzy Databases and Information retrieval systems.

TEXT BOOKS

1. Fuzzy Logic with Engineering Applications – Timothy J. Ross, Third Edition, John Wiley and Sons, 2010
2. Fuzzy Sets and Fuzzy Logic: Theory and Applications - George J. Klir and Bo Yuan , Prentice Hall, 1995.

REFERENCES

1. Kenneth H. Rosen, Discrete Mathematics and Its Applications with Combinatorics and GraphTheory, Seventh Edition, MGH,2011
2. Trembly J.P and Manohar R, "Discrete Mathematical Structures with Applications to Computer Science", TataMc Graw Hill Pub. Co. Ltd., New Delhi,2003.
3. Bernard Kolman, Robert C. Busby, Sharan Cutler Ross, "Discrete Mathematical Structures", Pearson Education Pvt Ltd., New Delhi,2003
4. Kenneth H .Rosen, "Discrete Mathematics and its Applications", 5/e, TataMc Graw Hill Pub. Co. Ltd, New Delhi2003

COURSE PLAN

Module	Contents	No. of Hours
1	**Basic Fuzzy Set Theory** Introduction to Fuzzy Concepts – Case for imprecision- utility and limitations of Fuzzy Systems, Classical Sets – Properties, Operations, Fuzzy Sets – Properties, Operations, Classical Relations–Properties Operations –Cartesian Product, Composition, Fuzzy Relations – Properties, Operations, Cardinality, Fuzzy Cartesian Product, Fuzzy Composition	**9**
2	**Fuzzy Membership Functions** Tolerance and Equivalence Relations – Crisp, Tolerance and Equivalence Relations – Fuzzy, Similarity Methods – Cosine, Minmax, Fuzzy Membership Functions-Features, Fuzzification, Defuzzification to crisp sets – λ-cuts, Linguistic Hedges	**9**
3	**Fuzzification and Defuzzification Methods** Development of Membership Functions – Intuition, Inference, Development of Membership Functions – Rank Ordering, Development of Membership Functions – Inductive reasoning, Defuzzification – Max membership principle, weighted average method, mean max membership, Defuzzification – Centroid method, Defuzzification – Center of Sums, Center of Largest area, First/Last of maxima, Defuzzification - exercises	**9**
4	**Fuzzy Inference** Classical Logic-Propositional Logic, Classical Logic-Predicate Logic, Fuzzy Logic,Fuzzy Approximation based reasoning, Fuzzy Rule based systems, Multiple conjuctive and disjunctive antecedents, aggregation, Graphical Techniques for Interference, Illustration of Graphical Techniques for Inference, Fuzzy Inference-Excercises	**9**
5	**Fuzzy Applications** Fuzzy Control Systems, Illustration of Fuzzy Control Systems, Fuzzy Classification, Fuzzy Pattern Recognition, Fuzzy Systems and Neural Networks, Fuzzy Clustering, Fuzzy Databases, Fuzzy Information Retrieval Systems	**9**
		45

CS1U42F	EMBEDDED SYSTEMS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

The objective of this course is to familiarize learners with the technologies behind embedded computing systems. This course introduces and explains the role of different hardware, software, and firmware components involved in the design and development of embedded systems. It discusses how real time operating systems incorporate specific features to ensure timeliness of critical tasks. The course also aims to provide insights about the design followed in several real-world embedded devices and expose the recent trends in embedded system design to the students.

COURSE OUTCOMES:

After the completion of the course the student will be able to

CO1	Describe the characteristics of different hardware/software components of an embedded system.	Understand
CO2	Map the design of an embedded system to an appropriate computational model.	Understand
CO3	Recommend appropriate process synchronization / communication / scheduling mechanisms for specific system scenarios.	Analyze
CO4	Describe the role of real-time operating systems in embedded devices.	Understand
CO5	Make use of design strategies for developing real-world embedded systems.	Apply

SYLLABUS

Module – 1 (Introduction to Embedded Systems)

Embedded Systems – Definitions, Embedded Systems vs. General Computing Systems, History, Classification, Application Areas, Purpose. Building Blocks of a Typical Embedded System – System Core (Microprocessors, Microcontrollers, DSP, ASICs, PLDs), Memory (Different ROMs and RAMs), Sensors and Actuators, I/O Subsystem Interface, Communication Interface, Embedded Firmware, Other System Components (Reset and Brown-out Protection Circuits,Oscillator Unit, Real-Time Clock, Watchdog Timer), Printed Circuit Board. Embedded System Design Process – Requirements, Specification, Architecture Design, Designing Hardware and Software Components, System Integration.

Module - 2 (System Modeling and Hardware Software Co-Design)

Computational Models in Embedded Design – Data Flow Graph, Control Data Flow Graph, State Machine Model, Sequential Program Model, Concurrent Process Model, Object-Oriented Model. Hardware Software Co-Design **–** Traditional Embedded Development Cycle, History**,** Advantages of the Co-Design Methodology, The Co-Design Process, Fundamental Issues in Hardware Software Co-Design. Hardware software trade-offs.

Module - 3 (Real-Time Embedded System Design)

Prerequisite Topics: Operating System – Basics, Types. Basics of Tasks, Process and Threads. Multiprocessing and Multitasking. Task Scheduling – Non-Preemptive (FIFO, LIFO, SJF) and Preemptive (SRT, RR, Priority-based, Rate-based).

Task Communication – Shared Memory, Message Passing, Remote Procedure Call and Sockets. Task Synchronization – Synchronization Issues – Race Condition, Deadlock, Priority Inversion, Priority Inheritance, Priority Ceiling. Synchronization Techniques – Spin Lock, Sleep & Wakeup, Semaphores. Selection of an RTOS for an Embedded Design – Functional and Non-Functional Requirements.

Module 4 – (Embedded Firmware Design and Development, and EDLC)

Embedded Firmware Design and Development – Firmware Design Approaches, Firmware Development Languages. Integration of Embedded Hardware and Firmware.
Embedded Product Development Life Cycle – Objectives, Different Phases, Modeling Techniques – Waterfall Model, Incremental Model, Evolutionary Model, Spiral Model.

Module 5 (Embedded System Industry – Case Studies and Applications)

Design Case Studies – Battery Operated Smart Card Reader, Automated Meter Reading System, Smart Watch.
Automotive and Aerospace Systems – Networked Control Systems in Cars and Airplanes, Vehicular Networks – CAN bus, Time-triggered Architecture, FlexRay and LIN.
Internet of Things Systems – IoT System Architectures - Use Cases (Smart Appliance, Monitoring and Control Systems). Networks for IoT – Networking concepts, Bluetooth, Bluetooth Low Energy, 802.15.4, ZigBee and WiFi. Databases and Timewheels. Smart Home Example.

TEXT BOOKS

1. K. V. Shibu, *Introduction to Embedded Systems*, McGraw Hill Education, Second Edition, 2017.
2. James K. Peckol, *Embedded Systems: A Contemporary Design Tool*, John Wiley & Sons, Second Edition, 2019.
3. Marilyn Wolf, *Computers as Components-Principles of Embedded Computing System Design*, Morgan Kaufmann, Elsevier, Fourth Edition, 2016.

REFERENCES

1. Jorgen Staunstrup and Wayne Wolf, *Hardware/Software Co-Design: Principles and Practice*, Springer Science & Business Media, 2013.
2. Raj Kamal, *Embedded Systems: Architecture, Programming and Design*. Tata McGraw-Hill Education, 2011.
3. Daniel D. Gajski, Samar Abdi, Andreas Gerstlauer, and Gunar Schirner, *Embedded System Design: Modeling, Synthesis and Verification*, Springer Science & Business Media, 2009.
4. Peter Marwedel, *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things*, Springer, 2017.

COURSE PLAN

Module	Contents	No.of Hours
1	**Introduction to Embedded Systems** Introduction–Embedded Systems, Characteristics and Quality Attributes of Embedded Systems, System Core (Microprocessors, Microcontrollers, DSP, ASICs, PLDs), System Core (Microprocessors, Microcontrollers, DSP, ASICs, PLDs),Memory (Different ROMs and RAMs), Sensors and Actuators, I/O Subsystem Interface ,Communication Interface, Embedded Firmware, Other System Components (Reset and Brown-out Protection Circuits, Oscillator Unit, Real-Time Clock, Watchdog Timer), Printed Circuit Board, Embedded System Design Process – Requirements, Specification, Architecture Design, Embedded System Design Process– Designing Hardware and Software Components, System Integration.	9
2	**System Modeling and Hardware Software Co-Design** Computational Models in Embedded Design – Data Flow Graph, Control Data Flow Graph , Computational Models in Embedded Design – State Machine Model, Sequential Program Model, Computational Models in Embedded Design – Concurrent Process Model, Object-Oriented Model. , Hardware Software Co-Design – Traditional Embedded Development Cycle, History, Advantages of the Co-Design Methodology , The Co-Design Process , Fundamental Issues in Hardware Software Co-Design. Hardware software trade-offs.	9
3	**Real-Time Embedded System Design** Task Communication–Shared Memory, Message Passing, Task Communication–Remote Procedure Call and Sockets, Task Synchronization–Synchronization Issues – Race Condition, Deadlock, Task Synchronization–Synchronization Issues – Priority Inversion, Priority Inheritance, Priority Ceiling, Synchronization Techniques – Spin Lock, Sleep & Wakeup, Semaphores, Synchronization Techniques – Spin Lock, Sleep & Wakeup, Semaphores, Selection of an RTOS for an Embedded Design – Functional and NonFunctional Requirements	9
4	**Embedded Firmware Design and Development, and EDLC** Firmware Design Approaches, Firmware Development Languages, Firmware Development Languages, Integration of Embedded Hardware and Firmware, Embedded Product Development Life Cycle–Objectives, Different Phases, Embedded Product Development Life Cycle – Modeling Techniques – Waterfall Model, Incremental Model, Evolutionary Model, Spiral Model (Review Only)	9
5	**Embedded System Industry – Case Studies and Applications** gn Case Studies–Battery Operated Smart Card Reader, Design Case ies–Automated Meter Reading System, Design Case Studies–Smart Watch, otive and Aerospace Systems – Networked Control Systems in Cars and lanes, Automotive and Aerospace Systems – Vehicular Networks – CAN Time-triggered Architecture, FlexRay and LIN, Internet of Things Systems T System Architectures – Use Cases (Smart Appliance, Monitoring and trol Systems), Internet of Things Systems – Networks for IoT – Networking epts, Bluetooth, Bluetooth Low Energy, 802.15.4, ZigBee and WiFi., net of Things Systems – Databases and Timewheels, Smart Home Example	9
		45

CS1U42G	COMPUTER VISION	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs. The curriculum covers the basics of image formation, key computer vision concepts, methods, techniques, pattern recognition, various problems in designing computer vision and object recognition systems. This course enables the learners to understand the fundamentals of computer vision and develop applications in computer vision.

COURSE OUTCOMES:

After the completion of the course, the student will be able to

CO1	Summarize basic concepts, terminology, theories, models and methods in the field of computer vision.	Understand
CO2	Explain basic methods of computer vision related to multi-scale representation, edge detection, detection of other primitives, stereo, motion and object recognition.	Understand
CO3	Describe principles of Segmentation, Motion Segmentation and Classification.	Understand
CO4	Select appropriate object Tracking and detection methods for computer vision applications	Understand
CO5	Implement a computer vision system for a specific problem	Apply

Syllabus

Module – 1 (Image Formation and Filtering)

Geometric Camera Models - Pinhole perspective, Intrinsic and Extrinsic Parameters, Geometric Camera Calibration. Linear Filters- Linear Filters and Convolution, Shift Invariant Linear Systems. Filters as Templates - Normalized Correlation and Finding Patterns.

Module - 2(Local Image Features and Stereo Vision)

Image Gradients - Computing the Image Gradient, Gradient Based Edge and Corner Detection. Stereopsis- Binocular Camera Geometry, Epipolar Constraint, Binocular Reconstruction, Local Methods for Binocular Fusion, Global Methods for Binocular Fusion.

Module - 3 (Segmentation)

Segmentation - Background subtraction, Interactive segmentation, Forming image regions. Segmentation by clustering - Watershed Algorithm. Motion Segmentation by Parameter Estimation- Optical Flow and Motion, Flow Models, Motion Segmentation with Layers.

Module- 4 (Classification and Tracking)

Classification - Classification Basics, Two-class and Multiclass classifiers, Error, Overfitting and Regularization, Cross Validation, Classifying Images of Single Objects.Tracking - Tracking Basics, Simple Tracking Strategies, Tracking by detection, Tracking Linear Dynamical models with Kalman filters.

Module - 5 (Finding Objects and other Applications)

Object detection - The Sliding Window Method. Object Recognition -Goals of Object Recognition System. Applications - Robot Navigation by stereo vision, Face detection, Face recognition, Activity Recognition, Tracking people.

TEXT BOOKS

1. David, and Jean Ponce. Computer vision: A modern approach. Prentice hall, 2011.

REFERENCES

1. Szeliski, Richard, Computer vision: algorithms and applications. Springer Science & Business Media, 2010.
2. Medioni, Gerard, Emerging topics in computer vision. and Sing Bing Kang. Prentice Hall PTR, 2004.
3. Trucco, Emanuele, and Alessandro Verri, Introductory techniques for 3-D computer vision. Vol. 201. Englewood Cliffs: Prentice Hall, 1998.
4. Faugeras, Olivier, and Olivier Autor Faugeras, Three-dimensional computer vision: a geometric viewpoint. MIT press, 1993.

COURSE PLAN

Module	Contents	No. of Hours
1	Geometric Camera model - Pinhole perspective, Geometric Camera model - Intrinsic Parameters, Geometric Camera model - Extrinsic Parameters, Geometric Camera Calibration – Linear Approach, Linear Filters and Convolution, Shift Invariant Linear Systems - Discrete convolution, Normalized Correlation and Finding patterns	9
2	Local Image Features - Computing the Image Gradient, Gradient Based Edge Detection, Gradient Based Corner Detection, Stereopsis - Binocular Camera Geometry and Epipolar Constraint, Essential Matrix and Fundamental Matrix, Binocular Reconstruction, Local Methods for Binocular Fusion, Global Methods for Binocular Fusion	9
3	Segmentation basics, Applications - Background Subtraction, Interactive Segmentation, Forming Image Regions, Segmentation by clustering - The Watershed Algorithm, Motion Segmentation by Parameter Estimation - Optical Flow and Motion, Flow Models and Motion Segmentation with Layers	9
4	Classification Basics, Two-class and Multiclass classifier, Error, Overfitting and Regularization, Cross Validation, Classifying Images of Single Objects Tracking Basics, Simple Tracking Strategies, Tracking by detection, Linear Dynamical models, The Kalman Filter background, Kalman filter algorithm	9
5	Detecting Objects in Images- The Sliding Window Method, Object Recognition - Goals of Object Recognition System, Application of binocular stereo vision - Robot Navigation, Face detection, Face recognition, Activity recognition, Tracking people	9
	Total hours	**45**

CS1U43A	FORMAL METHODS AND TOOLS IN SOFTWARE ENGINEERING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

The course enables the learners to apply formal methods for modelling, validation and verification of software systems. It covers a series of advanced tools that address challenges faced in design, coding and verification. This includes both an introduction to the theoretical underpinnings of these tools, as well as hands-on exploration.

COURSE OUTCOMES

After the completion of the course, the student will be able to

CO1	Explain the need and use of formal methods and tools in software engineering.	Understand
CO2	Demonstrate conceptual modelling of systems using *Alloy*.	Apply
CO3	Illustrate the process of proving correctness of code using Hoare-Triple based weakest precondition analysis.	Apply
CO4	Demonstrate program verification using VCC.	Apply

SYLLABUS

Module – 1 (Introduction)

Stages in software development; software defects –causes of software defects; techniques for dealing with software Defects-Testing and verification, formal methods and tools.

Module– 2 (Ensuring reliability in the design phase)

Conceptual modelling, the tool Alloy, conceptual modelling in Alloy, Analysing Alloy models, Fixing bugs in modelling, How Alloy works? Show that the Konigsberg Bridge Problem has no solution.

Module - 3 (Verification by Model Checking)

Verifier for Concurrent C (VCC): a Hoare-Triple- based tool for Verifying Concurrent C, intra-procedure verification of programs, ghost statements.

Module–4 (Program Verification)

Inter-procedure verification of programs in VCC, function contracts, pure functions, loop-invariants, proving total correctness of programs in VCC.

Module–5 (Ghost Language and Ownership in VCC)

Ghost Language of VCC, modelling programs in the ghost language, verification of a C program with respect to a ghost model, ownerships in VCC, Refinement for proving correctness, Proving refinements in VCC, Example problems

TEXT BOOKS

1. Daniel Jackson, Software Abstractions, MIT Press, 2011.

REFERENCES

1. Tutorial for Alloy Analyzer 4.0
2. E. Cohen, M. A., Hillebrand, S. Tobies, M. Moskal, W. Schulte, Verifying C Programs: A

VCC Tutorial, Working draft, version 0.2, July 10, 2015

3. The VCC Manual, Working draft, version 0.2, April 7, 2016.

COURSE PLAN

Module	Contents	No. of Hours
1	**Introduction** Stages in software development, Software defects and causes of software defects, Techniques for dealing with software defects, Testing and verification, Formal methods and formal verification	**9**
2	**Conceptual Modelling in Alloy** Introduction to Conceptual modelling, Overview of Alloy, Architecture of alloy, Conceptual modelling in Alloy, Analysing Alloy models, Fixing bugs in modelling, How Alloy works?, Show that the Konigsberg Bridge Problem has no solution	**9**
3	**Hoare Logic and Introduction to VCC** Introduction to VCC, Verifying C programs in VCC- Assertions, Logical Operators and Quantifiers, Assumptions, Overflows and unchecked arithmetic, Hoare Logic - Simple Imperative Programming Language, Partial Correctness Specification, Meaning of Hoare Triples, Hoare-Triple- based tool for Verifying Concurrent C, Partial vs. Total Correctness, Proving Partial Correctness, Inference Rules for the Simple Imperative Programming Language (Lecture 1), Inference Rules for the Simple Imperative Programming Language (Lecture 2), Weakest Precondition, Invariant vs. Inductive Invariant, Intra-procedure verification of programs, Verification of Hoare Triples	**9**
4	**Program Verification** Inter-procedure verification of programs in VCC, Function contracts, Pure functions, Quantifiers, loop-invariants and Object invariant, Triggers in VCC, Proving total correctness of programs in VCC	**9**
5	**Ghost Language and Ownership in VCC** Ghost Language of VCC, Modelling programs in the ghost language, Verification of a C program with respect to a ghost model, Ownerships in VCC, Phrasing refinement conditions in VCC, Proving refinements in VCC, Example problems (Lecture 1), Proving refinements in VCC, Example problems (Lecture 2)	**9**
		45

CS1U43B	CLIENT SERVER ARCHITECTURE	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

The syllabus is prepared with the view of preparing the Engineering Graduates to build effective Client/Server applications. This course aims at providing a foundation in decentralized computer systems, using the client/server model. The course content is decided to cover the essential fundamentals which can be taught within the given slots in the curriculum.

COURSE OUTCOMES

After the completion of the course the student will be able to

CO 1	Explain the basics of client/server systems and the driving force behind the development of client/server systems	**Understand**
CO 2	Outline the architecture and classifications of client/server systems	**Understand**
CO 3	Choose the appropriate client/server network services for a typical application	**Understand**
CO 4	Describe management services and issues in network	**Understand**
CO 5	Compare and summarize the web extensions and choose appropriate web services standards for an application	**Understand**

SYLLABUS

Module – 1 (Introduction)

Introduction to Client/Server computing - Driving forces behind Client/ Server, Client/ Server development tools, Development of client/server systems, Client/Server security, Organizational Expectations, Improving performance of client/server applications, Single system image, Downsizing and Rightsizing, Advantages of client server computing, Applications of Client/Server.

Module -2 (Client/Server Application Components)

Classification of Client/Server Systems- Two-Tier Computing, Middleware, Three-Tier Computing- Model View Controller (MVC), Principles behind Client/Server Systems. Client/Server Topologies. Existing Client/Server Architecture. Architecture for Business Information System.

Module -3 (Client/Server Network)

Client- Services, Request for services, RPC, Windows services, Print services, Remote boot services, other remote services, Utility Services. Dynamic Data Exchange (DDE). Object Linking and Embedding (OLE). Common Object Request Broker Architecture (CORBA).

Server- Detailed server functionality, Network operating system, Available platforms, Server operating system.

Module -4 (Client/ Server Systems Development)

Services and Support- System administration, Availability, Reliability, Scalability, Observability, Agility, Serviceability. Software Distribution, Performance, Network management. Remote Systems Management- RDP, Telnet, SSH, Security. LAN and Network Management issues, Training, Connectivity, Communication interface technology, Interprocess communication, Wide area network technologies, Network Acquisition, PC-level processing unit, X-terminals, Server hardware.

Module -5 (Client/Server Technology and Web Services)

Web Services History. Web Server Technology- Web Server, Web Server Communication, Role of Java for Client/Server on Web. Web Services- MicroServices, APIs, API Gateway, Authentication of users/clients, Tokens/Keys for Authentication, Service Mesh, Message Queues, SaaS, Web Sockets.
Client/Server/Browser – Server Technology, Client/Server Technology and Web Applications, Balanced Computing and the Server's Changing Role. Thin client computing- Computing models-Comparison-Computing Environment.Future of client/ server Computing Enabling Technologies, Transformational system.

TEXT BOOKS

1. Patrick Smith & Steave Guengerich, "Client / Server Computing", PHI
2. Dawna Travis Dewire, "Client/Server Computing", TMH

Module	Contents	No. of Hours
1	**Introduction** Driving forces behind Client/ Server, Client Server development tools, Development of client/server systems, Client/Server security, Organizational Expectations, Improving performance of client/server applications, Single system image, Downsizing and Rightsizing, Advantages and Applications of client server computing	9
2	**Client/Server Application Components** Classification of Client/Server Systems, Open System Standards, Two-Tier Computing, Three-Tier Computing, Middleware, Principles behind Client/Server Systems, Client/Server Topologies, Existing Client/Server Architecture, Architecture for Business Information System.	9
3	**Client/Server Network** The client: Services, Request for services, RPC, Windows services, Print services, Remote boot services, Utility Services & Other Services, Dynamic Data Exchange (DDE), Object Linking and Embedding (OLE), Common Object Request Broker Architecture (CORBA), The server: Detailed server functionality, the network operating system, Available platforms, the server operating system	9
4	**Client Server Systems Development** Services and Support, System administration, Availability, Reliability, Scalability, Observability, Agility Serviceability, Software Distribution,	9

	Performance, Network management, Remote Systems Management, RDP,Telnet,SSH, Security ,LAN and Network Management issues, Training, Connectivity, Communication interface technology, Interposes communication, wide area network technologies, Network Acquisition, PC-level processing unit, x-terminals, server Hardware	
5	**Client/Server Technology And Web Services** Web Services History , Web Server Technology , Web Server, Web Server Communication , Role of Java for Client/Server on Web, Web Services , MicroServices, APIs, API Gateway, Authentication of users/clients, Tokens/Keys for Authentication ,Service Mesh, Message Queues, SaaS, Web Sockets ,Client/Server Technology and Web Applications, Balanced Computing and the Server's Changing Role ,Thin client computing , Computing models, Computing Environment, Future of client/ server Computing Enabling Technologies, Transformational system	**9**

CS1U43C	PARALLEL COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

This course helps the learners to understand basic and advanced concepts of parallel computing. It covers Principles of Parallel Algorithm Design, Communication operations, Programming Using the Message Passing Paradigm, Programming Shared Address Space Platforms Thread Basics, and GPU Programming. This course enables a learner to design solutions to complex real world problems using parallel computing paradigms including thread parallelism, shared memory program, message passing interfaces, and vector processing.

COURSE OUTCOMES

CO1	Summarize the key parallel computational models	Understand
CO2	Appreciate and apply parallel and distributed algorithms in problem Solving	Apply
CO3	Appreciate the communication models for parallel algorithm development	Understand
CO4	Develop parallel algorithms using message passing paradigm	Apply
CO5	Formulate parallel algorithms for shared memory architectures.	Apply
CO6	Demonstrate the fundamental skills of heterogeneous computing with GPUs	Apply

SYLLABUS

Module- 1 (Principles of Parallel Algorithm Design)

Basic Introduction to Parallel Processing platforms. Preliminaries, Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing, Methods for Containing Interaction Overheads, Parallel Algorithm Models.

Module- 2 (Communication Operations)

Basic Communication Operations - One-to-All Broadcast and All-to-One Reduction, All-to-All Broadcast and Reduction, All-Reduce and Prefix-Sum Operations, Scatter and Gather, All-to-All Personalized Communication, Circular Shift, Improving the Speed of Some Communication Operation

Module-3 (Programming Using the Message Passing Paradigm)

Principles of Message-Passing Programming, The Building Blocks: Send and Receive Operations, MPI: The Message Passing Interface, Overlapping Communication with Computation, Collective Communication and Computation Operations, Groups and

Communicators.

Module 4 (Programming Shared Address Space Platforms Thread Basics)

Thread Basics, Why Threads? The POSIX Thread Application Programme Interface, Synchronization Primitives in POSIX, Controlling Thread and Synchronization Attributes, Thread Cancellation, Composite Synchronization Constructs, OpenMP: a Standard for Directive Based Parallel Programming, Specifying Concurrent Tasks in OpenMP, Synchronization Constructs in OpenMP, Data Handling in OpenMP, OpenMP Library Functions, OpenMP Applications: Parallel algorithm development for Matrix multiplication

Module 5 (GPU Programming)

Heterogeneous Parallel Computing, Architecture of a Modern GPU, Speeding up Real Applications, Data parallel computing, CUDA C Program Structure, A Vector Addition Kernel, Device Global Memory and Data Transfer, Kernel Functions and Threading, Kernel Launch, CUDA Thread Organization, Mapping Threads to Multidimensional Data, Synchronization and Transparent Scalability, Resource Assignment, Querying Device Properties, Thread Scheduling and Latency Tolerance, Importance of Memory Access Efficiency, Cuda Memory Types, Tiling for Reduced Memory Traffic, Tiled Matrix Multiplication Kernel, Boundary Checks

TEXT BOOKS

1. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, 2nd Ed, Addison-Wesley, 2003
2. David B. Kirk, Wen-mei W. Hwu, Programming Massively Parallel Processors: A Hands-on Approach, 3rd Ed., Morgan Kaufman, 2016.

REFERENCES

1. Steven Brawer, Introduction to Parallel Computing, Academic Press, (1989)

2. Barbara Chapman, Gabriele Jost, Ruud van der Pas, Using OpenMP: Portable Shared Memory Paralwlel Programming, MIT Press, 2008.

3. William Gropp, Ewing Lusk, Anthony Skjellum Using MPI: Portable Parallel Programming with the Message-Passing Interface, 3rd Ed, MIT Press, 2014.

4. Thomas Rauber, Gudula Rünger, Parallel Programming for Multicore and Cluster Systems, Springer, 2010

COURSE PLAN

Module	Contents	No.of Hours
1	**Basic Introduction to Parallel Processing** Basic Introduction to Parallel Processing platforms. Preliminaries, Decomposition Techniques – Recursive, Data, Decomposition Techniques – Exploratory, Speculative, Hybrid, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing – Static, Mapping Techniques for Load Balancing – Dynamic, Methods for Containing Interaction Overheads, Parallel Algorithm Models.	**9**
2	**Basic Communication Operations** One-to-All Broadcast and All-to-One Reduction, All-to-All Broadcast and Reduction, All-Reduce and Prefix-Sum Operations, Scallter Gather, All-to-All Personalized Communication, Circular Shift, Improving the Speed of Some Communication Operation	**9**
3	**Programming Using the Message Passing Paradigm** Principles of Message-Passing Programming, The Building Blocks: Send and Receive Operations, MPI: The Message Passing Interface, MPI: The Message Passing Interface : Illustration, Overlapping Communication with Computation, Overlapping Communication with Computation : Illustration, Collective Communication and Computation Operations, Collective Communication and Computation Operations : Illustration	**9**
4	**Programming Shared Address Space Platforms** Thread Basics, Why Threads? The POSIX Thread API, Synchronization Primitives in POSIX, Controlling Thread and Synchronization Attributes, Thread Cancellation, Composite Synchronization Constructs, OpenMP: a Standard for Directive Based Parallel Programming, Specifying Concurrent Tasks in OpenMP, Synchronization Constructs in OpenMP, Data Handling in OpenMP, OpenMP Library Functions, OpenMP Applications: Parallel algorithm development for Matrix multiplication	**9**
5	**GPU Programming**	**9**

	Heterogeneous Parallel Computing, Architecture of a Modern GPU, Speeding up Real Applications, Data parallel computing – CUDA C Program Structure, Vector Addition Kernel, Device Global Memory and Data Transfer, Kernel Functions and Threading, Kernel Launch, CUDA Thread Organization, Mapping Threads to Multidimensional Data, Synchronization and Transparent Scalability, Resource Assignment, Querying Device Properties, Thread Scheduling and Latency Tolerance, Importance of Memory Access Efficiency, Cuda Memory Types, Tiling for Reduced Memory Traffic, Tiled Matrix Multiplication Kernel, Boundary Checks	
		45

CS1U43D	DATA COMPRESSION TECHNIQUES	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

This course helps the learners to understand compression techniques on text, image, audio and video data. It covers lossy &lossless compression, RLE, JPEG, MPEG and its variants. This course enables the students to develop and implement compression algorithms on different domains.

`

COURSE OUTCOMES

After the completion of the course the student will be able to

CO1	Describe the fundamental principles of data compression.	Understand
CO2	Make use of statistical and dictionary based compression techniques for various applications	Apply
CO3	Illustrate various image compression standards.	Apply
CO4	Summarize video compression mechanisms to reduce the redundancy in video.	Understand
CO5	Use the fundamental properties of digital audio to compress audio data.	Understand

SYLLABUS

Module-1 (Modelling and types of compression)) 1

Introduction to Compression Techniques- Lossy compression & Lossless compression, Measures of Performance, Modeling and coding. Mathematical modelling for Lossless and lossy compression - Physical models and probability models.

Module – 2 (Basic Compression Methods)

Basic Compression Technique- Run length encoding, RLE Text compression. Statistical Methods-Prefix Codes, Binary Huffman coding, non-binary Huffman Algorithms, Arithmetic Coding.

Module - 3 (Text & Image Compression)

Dictionary based Coding- LZ77, LZ78 and LZW compression.Image Compression- Image standards, JPEG image Compression- Baseline JPEG, JPEG-LS.

Module - 4 (Video Compression)

Video Compression- Analog video, Digital Video, Motion Compensation. MPEG standards-MPEG 1, MPEG 4

Module - 5 (Audio Compression)

Audio Compression- Basics of Digital Audio, Basic Audio Compression Techniques, MPEG Audio Compression-Layer 1 coding, Layer 2 coding and Layer 3 coding.

TEXT BOOKS

1. David Solomon, Data compression: the complete reference, 4/e, Springer, January 2007
2. Khalid Sayood, Introduction to data compression, Morgan Kaufmann Publishers,2003.

REFERENCES

1) Stephen Welstead, Fractal and wavelet Image Compression techniques, PHI, 1999.
2) Sleinreitz, Multimedia System, Addison Wesley.
3) Mark Nelson and Jean-loup Gailly, The Data Compression Book, M&T Books.

COURSE PLAN

Module	Contents	No.of Hours
1	**Modelling and types of compression** Introduction to Compression Techniques- Lossy compression & Lossless compression, Measures of Performance, Modelling and coding, Physical model for lossless compression, Physical model for lossy compression, Probability model for lossless compression, Probability model for lossly compression	9
2	**Basic Compression Methods** Run length encoding, RLE Text compression, Statistical methods-Prefix Codes, Binary Huffman coding, Illustration of Binary Huffman coding, Non-binary Huffman Algorithms, Arithmetic Coding algorithm, Illustration of Arithmetic Coding algorithm,	9
3	**Text & Image Compression** LZ77 compression, LZ78 Compression, LZW Compression, Basics of Image compression and Image standards, Baseline JPEG Image compression, JPEG-LS Image compression	9
4	**Video Compression** Basics of Video Compression- Analog video and Digital Video, Motion Compensation, MPEG-1 standard and Video Syntax, MPEG-1 Pel Reconstruction, MPEG-4 standard, Functionalities for MPEG-4	9
5	**Audio Compression** Basics of Audio Compression, Digital Audio, Basic Audio Compression Techniques, MPEG Audio Compression basics-Frequency Domain Coding, Encoding: Layers I and II, Encoding: Layer II -Psychoacoustic Models, Psychoacoustic Models - Encoding: Layer III	9
		45

CS1U43F	DATA MINING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

This course helps the learner to understand the concepts of data mining and data warehousing. It covers the key processes of data mining, data preprocessing techniques, fundamentals and advanced concepts of classification, clustering, association rule mining, web mining and text mining. It enables the learners to develop new data mining algorithms and apply the existing algorithms in real-world scenarios.

COURSE OUTCOMES

After the completion of the course the student will be able to

CO1	Employ the key process of data mining and data warehousing concepts in application domains.	Understand
CO2	Make use of appropriate preprocessing techniques to convert raw data into suitable format for practical data mining tasks	Apply
CO3	Illustrate the use of classification and clustering algorithms in various application domains	Apply
CO4	Comprehend the use of association rule mining techniques.	Apply
CO5	Explain advanced data mining concepts and their applications in emerging domains	Understand

SYLLABUS

Module – 1 (Introduction to Data Mining and Data Warehousing)
Data warehouse-Differences between Operational Database Systems and Data Warehouses, Multidimensional data model- Warehouse schema, OLAP Operations, Data Warehouse Architecture, Data Warehousing to Data Mining, Data Mining Concepts and Applications, Knowledge Discovery in Database Vs Data mining, Architecture of typical data mining system, Data Mining Functionalities, Data Mining Issues.

Module - 2 (Data Preprocessing)
Data Preprocessing-Need of data preprocessing, Data Cleaning- Missing values, Noisy data, Data Integration and Transformation, Data Reduction-Data cube aggregation, Attribute subset selection, Dimensionality reduction, Numerosity reduction, Discretization and concept hierarchy generation.

Module - 3 (Advanced classification and Cluster analysis)
Classification- Introduction, Decision tree construction principle, Splitting indices -Information Gain, Gini indexDecision tree construction algorithms-ID3, Decision tree construction with presorting-SLIQ, Classification Accuracy-Precision, Recall.

Introduction to clustering-Clustering Paradigms, Partitioning Algorithm- PAM, Hierarchical Clustering-DBSCAN, Categorical Clustering-ROCK

Module 4: (Association Rule Analysis)
Association Rules-Introduction, Methods to discover Association rules, Apriori(Level-wise algorithm), Partition Algorithm, Pincer Search Algorithm, Dynamic Itemset Counting Algorithm, FP-tree Growth Algorithm.

Module 5 (Advanced Data Mining Techniques)
Web Mining - Web Content Mining, Web Structure Mining- Page Rank, Clever, Web Usage Mining- Preprocessing, Data structures, Pattern Discovery, Pattern Analysis. Text Mining-Text Data Analysis and information Retrieval, Basic measures for Text retrieval, Text Retrieval methods, Text Indexing Techniques, Query Processing Techniques.

TEXT BOOKS

1. Dunham M H, "Data Mining: Introductory and Advanced Topics", Pearson Education, New Delhi, 2003.
2. Arun K Pujari, "Data Mining Techniques", Universities Press Private Limited,2008.
3. Jaiwei Han and Micheline Kamber, "Data Mining Concepts and Techniques", Elsevier, 2006

REFERENCES

1. M Sudeep Elayidom, "Data Mining and Warehousing", 1st Edition, 2015, Cengage Learning India Pvt. Ltd.
2. MehmedKantardzic, "Data Mining Concepts, Methods and Algorithms", John Wiley and Sons, USA, 2003.
3. Pang-Ning Tan and Michael Steinbach, "Introduction to Data Mining", Addison Wesley, 2006.

COURSE PLAN

Module	Contents	No.of Hours
1	**Introduction to Data Mining and Data Warehousing** Data warehouse-Differences between Operational Database Systems and Data Warehouses, Multidimensional data model- Warehouse schema, OLAP Operations, DataWarehouse Architecture, Data Warehousing to Data Mining, Datamining Concepts and Applications, Knowledge Discovery in Database Vs Data mining, Architecture of typical data mining system,Data Mining Functionalities, Data Mining Functionalities, Data Mining Issues	9
2	**Data Preprocessing** Data Preprocessing: Need of Data Preprocessing, Data Cleaning- Missing values, Noisy data, Data integration, Data transformation, Data Reduction-Data cube aggregation, Attribute subset selection, Data Reduction-Dimensionality reduction, Numerosity reduction, Discretization and concept hierarchy generation	9
3	**Advanced classification and Cluster analysis** Classification- Introduction, Decision tree construction principle, Splitting indices-Information Gain, Gini index, Decision Tree- ID3, Decision Tree- ID3, Decision tree construction with presorting- SLIQ,	9

	Accuracy and error measures, evaluation, Introduction to clustering, Clustering Paradigms, Partitioning Algorithm- PAM, Hierarchical Clustering-DBSCAN, Categorical Clustering-ROCK	
4	**Association Rule Analysis** Association Rules: Introduction, Methods to discover association rules, A priori algorithm (Level-wise algorithm), A priori algorithm (Level-wise algorithm), Partition Algorithm, Pincer Search Algorithm, Pincer Search Algorithm, Dynamic Itemset Counting Algorithm, FP-tree Growth Algorithm	**9**
5	**Advanced Data Mining Techniques** Web Mining - Web Content Mining, Web Structure Mining- Page Rank, Web Structure Mining –Clever algorithm, Web Usage Mining- Preprocessing, Data structures, Web Usage Mining -Pattern Discovery, Pattern Analysis, Text Mining-Text Data Analysis and information Retrieval, Basic measures for Text retrieval, Text Retrieval methods, Text Indexing Techniques Query Processing Techniques	**9**
		45

CS1U43G	MOBILE COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

The course is designed with the view of preparing the engineering students capable of understanding the communication protocols, various architectures and security features used in mobile computing. This course covers basics of mobile computing, architecture of wireless transmission systems and next generation networks. This course enables the learners to acquire advanced concepts on wireless communication systems and mobile ad-hoc networks.

COURSE OUTCOMES

After the completion of the course the student will be able to

CO#	Course Outcomes
CO1	Explain the various mobile computing applications, services, design considerations and architectures (**Cognitive knowledge: Understand**)
CO2	Describe the various technology trends for next generation cellular wireless networks and use the spreading concept on data transmission **(Cognitive knowledge: Apply)**
CO3	Summarize the architecture of various wireless LAN technologies **(Cognitive knowledge: Understand)**
CO4	Identify the functionalities of mobile network layer and transport layer **(Cognitive knowledge: Understand)**
CO5	Explain the features of Wireless Application Protocol **(Cognitive knowledge: Understand)**
CO6	Interpret the security issues in mobile computing and next generation technologies **(Cognitive knowledge: Understand)**

SYLLABUS

Module - 1 (Mobile Computing Basics)

Introduction to mobile computing – Functions, Middleware and Gateways, Application and services. Mobile computing architecture – Internet: The Ubiquitous network, Three-tier architecture for Mobile Computing, Design considerations for mobile computing.

Module – 2 (Wireless Transmission and Communication Systems)

Spread spectrum – Direct sequence, Frequency hopping. Medium Access Control – Space Division Multiple Access (SDMA), Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA). Satellite Systems – Basics, Applications, Geostationary Earth Orbit (GEO), Low Earth Orbit (LEO),

Medium Earth Orbit (MEO), Routing, Localization, Handover. Telecommunication Systems - Global System for Mobile Communication (GSM) services, Architecture, Handover, Security.

Module – 3 (Wireless LANs)

Wireless LAN - Advantages, Design goals, Applications, Infrastructure Vs Ad-hoc mode, IEEE 802.11 System Architecture, Protocol Architecture, Physical layer, Medium Access Control layer, HIPERLAN-1, Bluetooth.

Module – 4 (Mobile Network and Transport Layer)

Mobile network layer – Mobile Internet Protocol (IP), Dynamic Host Configuration Protocol (DHCP), Mobile ad-hoc networks – Routing, Dynamic Source Routing (DSR), Destination Sequence Distance Vector (DSDV), Ad-hoc routing protocols. Mobile transport layer – Traditional Transmission Control Protocol (TCP), Improvements in Classical TCP. Wireless Application Protocol (WAP) - Architecture, Wireless Datagram Protocol (WDP), Wireless Transport Layer Security (WTLS), Wireless Transaction Protocol (WTP), Wireless Session Protocol (WSP).

Module – 5 (Mobile Security and Next Generation Networks)

Security issues in mobile computing - Information security, Security techniques and algorithms, Security models. Next generation networks - Orthogonal Frequency Division Multiplexing (OFDM), Wireless Asynchronous Transfer Mode (WATM), Multi Protocol Label Switching (MPLS), 10 pillars of 5G, Security for 5G communication.

TEXT BOOKS

1. Asoke K. Talukder, Hasan Ahmad, Roopa R Yavagal, Mobile Computing Technology-Application and Service Creation, 2/e, McGraw Hill Education.
2. Jochen Schiller, Mobile Communications, Pearson Education Asia, 2008.
3. Jonathan Rodriguez, Fundamentals of 5G Mobile Networks, Wiley Publishers, 2015.

REFERENCES

1. Raj Kamal, Mobile Computing, 2/e, Oxford University Press.
2. Andrew S. Tanenbaum, Computer Networks, PHI, 3/e, 2003
3. Theodore S. Rappaport, Wireless Communications Principles and Practice, 2/e, PHI, New Delhi, 2004.
4. Curt M. White, Fundamentals of Networking and Communication 7/e, Cengage learning.

COURSE PLAN

Module	Contents	No.of Hours
1	**Mobile Computing Basics** Introduction to mobile computing – Functions, Middleware and Gateways, Application and services, Internet: The Ubiquitous network, Three-tier architecture for Mobile Computing, Design considerations for mobile computing	9
2	**Wireless Transmission and Communication Systems** Direct sequence spread spectrum, Frequency hopping spread spectrum,	9

	Space Division Multiple Access (SDMA), Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA), Satellite Systems Basics, Applications, Geostationary Earth Orbit (GEO), Low Earth Orbit (LEO), Medium Earth Orbit (MEO), Routing, Localization, Handover, Global System for Mobile Communication (GSM) services, Architecture, Handover, Security	
3	**Wireless LANs** Wireless LAN - Advantages, Design goals, Applications, Infrastructure Vs Ad-hoc mode, IEEE 802.11 System Architecture, Protocol Architecture, Physical layer, Medium Access Control layer, HIPERLAN-1, Bluetooth	9
4	**Mobile Network and Transport Layer** Mobile Internet Protocol (IP), Dynamic Host Configuration Protocol (DHCP), Mobile ad-hoc networks – Routing, Dynamic Source Routing (DSR), Destination Sequence Distance Vector (DSDV), Ad-hoc routing protocols, Traditional Transmission Control Protocol (TCP), Improvements in Classical TCP, Wireless Application Protocol (WAP) – Architecture, Wireless Datagram Protocol (WDP), Wireless Application Protocol (WAP) – Architecture, Wireless Datagram Protocol (WDP), Wireless Transport Layer Security (WTLS), Wireless Transaction Protocol (WTP), Wireless Session Protocol (WSP)	9
5	**Mobile Security and Next Generation Networks** Information security, Security techniques, Security algorithms, Security models, Introduction to Next generation networks, Orthogonal Frequency Division Multiplexing (OFDM), Wireless Asynchronous Transfer Mode (WATM), Multi Protocol Label Switching (MPLS), 10 pillars of 5G, Security for 5G communication.	9
		45

CS1U44A	HIGH PERFORMANCE COMPUTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

This course helps the learners to understand the different architectural features of high-end processors. This course discusses the Basics of high-end processors Architecture, Instruction-Level Parallelism, Data-Level Parallelism, Thread Level Parallelism, and GPU Architectures. This course enables the students to provide solutions to real-world problems making use of the capabilities of HPC systems.

COURSE OUTCOMES

After the completion of the course the student will be able to

CO1	Describe different types of modern processing environments and parallel computing hardware .	Understand
CO2	Summarize the concepts of Instruction Level Parallelism .	Understand
CO3	Appreciate the idea of Data Level Parallelism.	Apply
CO4	Demonstrate the concept of Thread Level Parallelism.	Apply
CO5	Describe the basics of GPU architecture.	Understand

SYLLABUS

Module-1 (Basics of Architecture)

Computers - Classes of Parallelism and Parallel Architectures – Defining Computer Architecture – Dependability – Quantitative Principles of Computer Design – Basics of Memory Hierarchies – Virtual Memory and Virtual Machines – Pipelining

Module-2 (Instruction-Level Parallelism)

Instruction-Level Parallelism: Concepts and Challenges – Basic Compiler Techniques for Exposing ILP – Reducing Branch Costs With Advanced Branch Prediction – Hardware-Based Speculation – Multithreading: Exploiting Thread-Level Parallelism to Improve Uniprocessor Throughput

Module-3 (Data-Level Parallelism)

Vector Architecture – SIMD Instruction Set Extensions for Multimedia – Graphics Processing Units – Detecting and Enhancing Loop-Level Parallelism

Module-4 (Thread Level Parallelism)

Multiprocessor Architecture: Issues and Approach – Centralized Shared-Memory Architectures –Performance of Symmetric Shared-Memory Multiprocessors– Distributed Shared-Memory and Directory-Based Coherence – Synchronization: The Basics – Introduction to Memory Consistency

Module-5 (GPU Architectures)

The CPU-GPU system as an accelerated computational platform – The GPU and the thread engine – Characteristics of GPU memory spaces – The PCI bus: CPU to GPU data transfer overhead – Multi-GPU platforms – Potential benefits of GPU – accelerated platforms.

TEXT BOOKS

1. John L. Hennessy, David A. Patterson Computer Architecture, Sixth Edition A Quantitative Approach, Morgan Kaufman, Fifth Edition, 2012.
2. Robert Robey, Yuliana Zamora, Parallel and High-Performance Computing, Manning Publications, First Edition, 2021.

REFERENCES

1. Thomas Sterling, Matthew Anderson, and MaciejBrodowicz, High-Performance Computing – Modern Systems and Practices, First Edition, 2017.
2. Charles Severance, Kevin Dowd,High-Performance Computing, O'Reilly Media, Second Edition, 1998.
3. Kai Hwang, Faye Alaye Briggs, Computer Architecture and Parallel Processing, McGraw-Hill, 1984.

COURSE PLAN

Module	Contents	No.of Hours
1	**Basics of Architecture** Classes of Computers, Classes of Computers, Classes of Parallelism and Parallel Architectures, Dependability, Quantitative Principles of Computer Design, Basics of Memory Hierarchies, Virtual Memory and Virtual Machines, Pipelining	7
2	**Introduction to Syntax Analysis** Instruction-Level Parallelism: Concepts and Challenges, Basic Compiler Techniques for Exposing ILP, Reducing Branch Costs With Advanced Branch Prediction, Hardware-Based Speculation, Multithreading, Exploiting Thread-Level Parallelism to Improve Uniprocessor Throughput – Lecture 1, Exploiting Thread-Level Parallelism to Improve Uniprocessor Throughput – Lecture 2.	9
3	**Data-Level Parallelism** Vector Architecture -Lecture 1, Vector Architecture -Lecture 2, SIMD Instruction Set Extensions for Multimedia – Lecture 1, SIMD Instruction Set Extensions for Multimedia – Lecture 2, Graphics Processing Units, Detecting and Enhancing Loop-Level Parallelism – Lecture 1, Detecting and Enhancing Loop-Level Parallelism – Lecture 2	9
4	**Thread Level Parallelism** Multiprocessor Architecture: Issues and Approach, Centralized Shared-	9

	Memory Architectures – Lecture 1, Centralized Shared-Memory Architectures – Lecture 2, Performance of Symmetric Shared-Memory Multiprocessors, Distributed Shared-Memory, Directory-Based Coherence, Synchronization, Introduction to Memory Consistency	
5	**GPU Architectures** The CPU-GPU system as an accelerated computational platform, The GPU and the thread engine – Lecture 1, The GPU and the thread engine – Lecture 2, Characteristics of GPU memory spaces, PCI bus: CPU to GPU data transfer overhead, Multi-GPU platforms, Potential benefits of GPU-accelerated platforms	**9**
		45

CS1U44B	BLOCKCHAIN TECHNOLOGIES	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

The purpose of this course is to create awareness and understanding among students on the foundation of blockchain technology. The course introduces the cryptographic principles behind blockchain and helps the students understand concepts like consensus, crypto-currency, smart contracts, use cases etc. The course enables students to develop simple decentralized applications using blockchain networks such as Ethereum.

COURSE OUTCOMES

After the completion of the course the student will be able to

CO1	Illustrate the cryptographic building blocks of blockchain technology.	Understand
CO2	Explain the fundamental concepts of blockchain technology.	Understand
CO3	Summarize the classification of consensus algorithms.	Understand
CO4	Explain the concepts of first decentralized cryptocurrency bitcoin.	Understand
CO5	Explain the use of smart contracts and its use cases.	Understand
CO6	Develop simple applications using Solidity language on Ethereum platform.	Apply

SYLLABUS

Module – 1 (Fundamentals of Cryptography)

Introduction to Cryptography, Symmetric cryptography – AES. Asymmetric cryptography – RSA. Elliptic curve cryptography, Digital signatures – RSA digital signature algorithms. Secure Hash Algorithms – SHA-256. Applications of cryptographic hash functions – Merkle trees, Distributed hash tables.

Module – 2 (Fundamentals of Blockchain Technology)

Blockchain – Definition, architecture, elements of blockchain, benefits and limitations, types of blockchain. Consensus – definition, types, consensus in blockchain.
Decentralization – Decentralization using blockchain, Methods of decentralization, Routes to decentralization, Blockchain and full ecosystem decentralization.

Module - 3 (Consensus Algorithms and Bitcoin)

Consensus Algorithms, Crash fault-tolerance (CFT) algorithms – Paxos, Raft. Byzantine fault-tolerance (BFT) algorithms – Practical Byzantine Fault Tolerance (PBFT), Proof of work (PoW), Proof of stake (PoS), Types of PoS.
Bitcoin – Definition, Cryptographic keys – Private keys, public keys, addresses. Transactions –Lifecycle, coinbase transactions, transaction validation. Blockchain – The genesis block.
Mining – Tasks of miners, mining algorithm, hash rate. Wallets – Types of wallets.

Module - 4 (Smart Contracts and Use cases)

Smart Contracts – Definition, Smart contract templates, Oracles, Types of oracles, Deploying smart contracts. Decentralization terminology – Decentralized applications, Decentralized Autonomous Organizations.
Use cases of Blockchain technology – Government, Health care, Finance, Supply chain management.
Blockchain and allied technologies – Blockchain and Cloud Computing, Blockchain and Artificial Intelligence.

Module - 5 (Ethereum and Solidity)

Ethereum – The Ethereum network. Components of the Ethereum ecosystem – Keys and addresses, Accounts, Transactions and messages. The Ethereum Virtual Machine, Blocks and blockchain.
The Solidity language – The layout of a Solidity source code, Structure of a smart contract, variables, data types, control structures, events, inheritance, libraries, functions, error handling.
Smart contracts Case study: Voting, Auction.

TEXT BOOKS

1. Imran Bashir, Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, DApps, cryptocurrencies, Ethereum, and more, Packt Publishing, Third edition, 2020.

REFERENCES

2. Ritesh Modi, Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain, Packt Publishing, First edition, 2018.
3. Kumar Saurabh, Ashutosh Saxena, Blockchain Technology: Concepts and Applications, First Edition, Wiley Publications, First edition, 2020.
4. Chandramouli Subramanian, Asha A George, et al, Blockchain Technology, Universities Press (India) Pvt. Ltd, First edition, August 2020.
5. Lorne Lantz, Daniel Cawrey, Mastering Blockchain: Unlocking the Power of Cryptocurrencies, Smart Contracts, and Decentralized Applications, O'Reilly Media, First edition, 2020.
6. Andreas M. Antonopoulos, Gavin Wood, Mastering Ethereum: Building Smart Contracts and DApps, O'Reilly Media, First edition, 2018.

COURSE PLAN

Module	Contents	No.of

		Hours
1	**Fundamentals of Cryptography** Introduction to cryptography, Symmetric cryptography, AES, Asymmetric cryptography, RSA, Elliptic curve cryptography, Digital signatures – RSA digital signature algorithm, Secure Hash Algorithms – SHA-256, Applications of cryptographic hash functions – Merkle trees, Distributed hash tables,	9
2	**Fundamentals of Blockchain Technology** Blockchain – definition and architecture, Elements of blockchain, Blockchain – benefits and limitations, types, Consensus – definition, types, consensus in blockchain, Decentralization using blockchain, Methods of decentralization, Routes to decentralization, Blockchain and full ecosystem decentralization	9
3	**Consensus Algorithms and Bitcoin** Consensus Algorithms – Crash fault-tolerance (CFT) algorithms – Paxos, Raft (working is expected), Byzantine fault-tolerance (BFT) algorithms – Practical Byzantine Fault Tolerance (PBFT) (working is expected), Proof of work (PoW), Proof of stake (PoS), Types of PoS, Bitcoin – Definition, Cryptographic keys – Private keys, public keys, addresses, Transactions – Lifecycle, coinbase transactions, transaction validation, Blockchain – The genesis block. Mining – Tasks of miners, Mining – mining algorithm, hash rate. Wallets – Types of wallets	9
4	**Smart Contracts and Use cases** Smart Contracts – Definition, Smart contract templates, Oracles, Types of oracles, Deploying smart contracts, Decentralization terminology – Decentralized applications, Decentralized Autonomous Organizations, Use cases of Blockchain technology – Government, Health care, Use cases of Blockchain technology – Finance, Supply chain management, Blockchain and Allied Technologies – Blockchain and Cloud Computing, Blockchain and Artificial Intelligence	9
5	**Ethereum and Solidity** Ethereum - The Ethereum network, Components of the Ethereum ecosystem – Keys and addresses, Accounts, Components of the Ethereum ecosystem – Transactions and messages, The Ethereum Virtual Machine, Ethereum Blocks and blockchain, The Solidity language – The layout of a Solidity source code, Structure of a smart contract, variables, data types, The Solidity language – control structures, events, inheritance, libraries, The Solidity language – functions, error handling, Smart contracts Case study: Voting, Smart contracts Case study: Auction	9
		45

CS1U44C	IMAGE PROCESSING TECHNIQUE	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

This course helps the learners understand the core concepts and applications of Digital Image Processing. It covers Digital Image Fundamentals, Image Transforms, Image Enhancement in Spatial and Frequency Domain, Image Restoration & Image Segmentation and Morphological Operations & Representation and Description. The learners will be able to develop new algorithms, tools, and application software for real-world applications involving image processing.

COURSE OUTCOMES : After the completion of the course, the student will be able to

CO1	Explain the concepts of image formation and the basis of digital image processing.	Understand
CO2	Demonstrate the role of image transforms in representing, highlighting, and modifying image features.	Apply
CO3	Solve image enhancement problems using spatial and frequency domain techniques.	Apply
CO4	Make use of the concept of image restoration and image segmentation techniques in real-world problems.	Apply

SYLLABUS

Module – 1 (Digital Image Fundamentals)

Elements of Visual Perception, A Simple Image Formation Model. Spatial and Intensity Resolution. Image Interpolation. Classification of Digital Images. Image Types. Image Storage Mechanisms. Arithmetic and Logical Operations. Geometric Spatial Transformations and Image Registration. Image File Formats. Colour Fundamentals and Colour Models.

Module - 2 (Image Transforms)

Basic concept of spatial domain and frequency domain, Unitary transform, Discrete Fourier Transform- 2D DFT, 4 order DFT Transform coefficients, Forward and inverse transform, Discrete Cosine Transform- 2D DCT, 4 order DCT Transform Coefficients(No derivation needed), Forward and Inverse DCT, Hadamard Transform.

Module - 3 (Image Enhancement in Spatial and Frequency Domain)

Point operations- Clipping and Thresholding, Digital Negative, Intensity Level Slicing, Bit

Extraction, Range Compression. Spatial Operations- Fundamentals of spatial convolution and correlation, Spatial averaging and spatial Low pass filtering, Directional Smoothing, Median Filtering, Unsharp masking and Crispening.

Basics of Filtering in Frequency Domain, Filters, Smoothing Frequency Domain Filters- Sharpening Frequency Domain Filters

Module - 4 (Image Restoration & Image Segmentation)

Image degradation model, Noise models, Mean Filters, Order Statistic filter, Adaptive filters. Edge Detection, gradient operators, Laplace operators and zero crossings. Thresholding, Basic Global Thresholding, Optimum global thresholding using Otsu method, Multiple thresholds, Variable thresholding, Multivariable thresholding. Region-Based Approach to Segmentation.

Module - 5 (Morphological Operations & Representation and Description)

Structuring Element, Dilation and Erosion, Opening and Closing, Hit or Miss Transformation.

Boundary Following. Chain Codes. Polygonal Approximation. Boundary Descriptors. Regional Descriptors. Relational Descriptors.

TEXT BOOKS

1. Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing (English) 3rd Edition, Pearson India, 2013
2. A K. Jain, Fundamentals of digital image processing, Prentice Hall of India, 1989.

REFERENCES

1. Al Bovik, The Essential Guide to Image Processing, Academic Press, 2009.
2. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing, Analysis, and Machine Vision, Thomson Learning, 2008.
3. S Jayaraman, S Esakkirajan and T Veerakumar, Digital Image Processing, McGraw Hill Education, 2009.

COURSE PLAN

Module	Contents	No. of Hours
1	**Digital Image Fundamentals** Elements of Visual Perception, A Simple Image Formation Model, Spatial and Intensity Resolution, Image Interpolation, Classification of Digital Image, Image Types, Image Storage Mechanisms, Arithmetic and Logical Operations, Geometric Spatial Transformations and Image Registration, Image File Formats, Colour Fundamentals and Colour Models	9
2	**Image Transforms** Basic concept of spatial domain and frequency domain, Need of Image Transform, Basic properties of unitary transform, Discrete Fourier transform, Proof DFT is Unitary, 4 order DFT Transform coefficients (Derivation), Problems (4 order DFT), Discrete Cosine Transform- 2D	9

	DCT, 4 order DCT Transform Coefficients(No derivation needed), Hadamard Transform	
3	**Image Enhancement in spatial and frequency domain** Point operations- Clipping and Thresholding, Digital Negative. Intensity Level Slicing, Bit Extraction, Range Compression + (Work out problems), Spatial Operations-Fundamentals of spatial convolution and correlation, Spatial averaging and spatial Low pass filtering, Directional Smoothing, Median Filtering, Unsharp masking and Crispening, Basics of Filtering in Frequency Domain, Smoothing Frequency Domain Filters : Ideal Low Pass Filter; Gaussian Low Pass Filter; Butterworth Low Pass Filter, Sharpening Frequency Domain Filters: Ideal High Pass Filter; Gaussian High Pass Filter; Butterworth High Pass filter	**9**
4	**Image Restoration & Image Segmentation** Image degradation model, Noise models, Mean Filters – Order Statistic filter – Adaptive filters, Edge Detection, Gradient operators, Laplace operators and zero crossings, Thresholding- Basic Global Thresholding, Optimum global thresholding using Otsu method, Multiple thresholds, Variable thresholding, Multivariable thresholding, Region-Based Approach to Segmentation	**9**
5	**Morphological Operations & Representation and Description** Structuring Element. Dilation and Erosion, Morphological Opening, Closing, Hit or Miss Transformation, Boundary Following. Chain Codes, Polygonal Approximation, Boundary Descriptors, Regional Descriptors, Relational Descriptors	**9**
		45

CS1U44D	INTERNET OF THINGS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

This course equips the learners with fundamental of the Internet of Things(IoT) and the IoT ecosystem. It covers the architecture of IoT, communication mechanisms, protocols, hardware, software, data analytics, and the cloud platforms for IoT. This course enables the students to design smart IoT applications for real world problems using Raspberry Pi.

COURSE OUTCOMES:

After the completion of the course the students will be able to

CO1	Outline the fundamentals of IoT and its underlying physical and logical architecture.	Understand
CO2	Explain the hardware architectures for IoT	Understand
CO3	Outline the Network architectures for IoT	Understand
CO4	Implement data analytics on the IoT platforms	Apply
CO5	Appreciate the security considerations in IoT	Understand
CO6	Implement IoT applications using the available hardware and software.	Apply

SYLLABUS

Module- 1 (IoT Architecture)

What is IoT, Genesis of IoT, IoT and Digitization, IoT Impact, Convergence of IT and IoT, IoT Challenges, IoT Network Architecture and Design, Drivers Behind New Network Architectures, Comparing IoT Architectures, A Simplified IoT Architecture, The Core IoT Functional Stack, IoT Data Management and Compute Stack.

Module- 2 (Engineering IoT Networks)

Smart Objects: The “Things” in IoT, Sensors, Actuators, and Smart Objects, Sensor Networks, Connecting Smart Objects, Communications Criteria, IoT Access Technologies

Module- 3 (IoT Network Layer)

IP as the IoT Network Layer, The Business Case for IP, The need for Optimization, Optimizing IP for IoT, Profiles and Compliances, Application Protocols for IoT, The Transport Layer, IoT Application Transport Methods

Module 4 (Data Analytics for IoT)

Data and Analytics for IoT, An Introduction to Data Analytics for IoT, Machine Learning, Big Data Analytics Tools and Technology, Edge Streaming Analytics, Network Analytics, Securing IoT, A Brief History of OT Security, Common Challenges in OT Security, Differences between IT and OT Security Practices and Systems, Formal Risk Analysis Structures: OCTAVE and FAIR.

Module 5 (Developing IoT Systems)

IoT Logical Design using Python, IoT Physical Devices and Endpoints - Raspberry Pi interfaces, Programming Raspberry Pi using Python, Other IoT devices, IoT Physical devices and Cloud offerings, Cloud Storage Models, WAMP - Autobahn for IoT, Django, Designing RESTful Web API, Cloud Web Services for IoT.

TEXT BOOKS

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", 1st Edition, Pearson Education (Cisco Press Indian Reprint).
2. Arshadeep Bahga, Vijay Madisetti, "Internet of Things: A hands-on approach", University Press, 2015 (First edition)

REFERENCES

1. Rajkamal, "Internet of Things: Architecture and Design Principles", McGraw Hill (India) Private Limited
2. Dieter Uckelmann, Mark Harrison, Michahelles Florian (Ed.), Architecting the internet of things, Springer, 2011
3. Dr. Ovidiu Vermesan, Dr. Peter Friess, Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, River Publishers, 2013
4. Simon Monk, "Programming Arduino: Getting Started with Sketches", McGraw Hill Publications

COURSE PLAN

Module	Contents	No. of Hours
1	**IoT Architecture** What is IoT, Genesis of IoT, IoT and Digitization, IoT Impact, Convergence of IT and IoT, IoT Challenges, IoT Network Architecture and Design, Drivers Behind New Network Architectures, Comparing IoT Architectures, A Simplified IoT Architecture, The Core IoT Functional Stack, IoT Data Management and Compute Stack	9
2	**Engineering IoT Networks** Smart Objects: The "Things" in IoT, Sensors, Actuators, and Smart	9

	Objects, Sensor Networks, Connecting Smart Objects, IoT Access Technologies –IEEE 802.15.4 (g/e), 1901.2a, IoT Access Technologies - 802.11ah, LoRaWAN, IoT Access Technologies – LoRaWAN, NBIoT, LTE	
3	**IoT Network Layer** IP as the IoT Network Layer, The Business Case for IP, The need for Optimizing IP for IoT, Optimizing IP for IoT, Profiles, and Compliance, Application Protocols for IoT – CoAP, Application Protocols for IoT – MQTT, The Transport Layer, IoT Application Transport Methods, The Transport Layer, IoT Application Transport Methods	**9**
4	**Data Analytics for IoT** An Introduction to Data Analytics for IoT, Machine Learning, Big Data Analytics Tools and Technology, Edge Streaming Analytics, Network Analytics, A Brief History of OT Security, Common Challenges in OT Security, Differences between IT and OT Security Practices and Systems, Formal Risk Analysis Structures: OCTAVE and FAIR	**9**
5	**Developing IoT Systems** IoT Logical Design using Python, IoT Physical Devices and Endpoints, Raspberry Pi interfaces, Programming Raspberry Pi using Python, Other IoT devices, Cloud Storage Models, WAMP-Autobahn for IoT, Django, Designing RESTful Web API, Cloud Web Services for IoT	**9**
		45

CS1U44E	SOFTWARE TESTING	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

This is a course in theoretical computer science that introduces the concepts and methods in software testing. It covers various techniques for test case design used to test software artifacts, including requirements, design, and code, the different techniques for test case design based on graphs, programming language syntaxes and symbolic execution using PEX tool. It enables the learners to follow a systematic software testing approaches while developing applications.

COURSE OUTCOMES: After the completion of the course the student will be able to:-

CO1	Illustrate a range of different software testing techniques and be able to apply specific unit testing method to the projects using Junit.	Understand
CO2	Make use of tools like Junit and Muclipse to perform mutation testing and also the different unit testing methods	Apply
CO3	Solve graph coverage criteria problems in terms of control flow graph and data flow graph for a given program.	Apply
CO4	Demonstrate the importance of black-box approaches in terms of domain and functional testing.	Understand
CO5	Illustrate different grey box testing techniques, parameterized unit testing and symbolic execution.	Understand

SYLLABUS

Module - 1 (Introduction to Software Testing)

Some Popular Errors – Ariane 5, Therac 25, Intel Pentium Bug. What is Software testing? Why should it be tested? Software Quality, Role of Testing. Testing Process - Level 0 thinking, Level 1 thinking, Level 2 thinking, Level 3 thinking, Level 4 thinking. Software Testing Terminologies - Verification, Validation and Testing, Faults, Error and Bug, Test cases, Coverage Criteria. Types of Testing- Unit testing, integration testing, System testing, Acceptance testing, Beta testing, Functional testing, Stress testing, Performance testing, Usability testing and Regression testing. Testing Methods - Black Box testing, White Box testing, Grey Box testing.

Module - 2 (Unit Testing)

Concept of Unit testing. Static Unit testing. Dynamic Unit testing - Control Flow testing, Data Flow testing, Domain testing, Functional Program testing. Mutation testing - Mutation and Mutants, Mutation operators, Mutation score. Junit - Framework for Unit testing. Case Study - Mutation testing using Junit and Muclipse.

Module - 3 (Unit Testing - White Box Approaches)

Overview of Graph Coverage Criteria. Structural Graph Coverage Criteria - Node/vertex coverage, Edge coverage, Edge pair coverage, Path coverage, Complete path coverage, Prime path coverage, Complete round trip coverage, Simple round trip coverage. Data Flow Criteria - du paths, du pairs. Subsumption Relationships among Graph Coverage Criteria. Graph Coverage for Source Code - Control flow graphs for code, CFG: If statement, CFG: If statement with return, CFG: Switch-case, CFG: Loops, CFG: Exceptions (try-catch). Example program – Statistics. Graph Coverage for Design Elements - Call graphs and classes, Class inheritance testing: Coverage criteria, Coverage criteria on inheritance graph, Data flow at the design level, Inter-procedural DU pairs, Coupling du-pairs example. Example - Quadratic Root. Case Study -Graph Based testing using JUnit Framework.

Module - 4 (Unit Testing - Black Box Approaches)

Domain Testing / Input Space Partitioning - Partitions of a set. Input domain modelling - Interface-based approach, Functionality-based approach. Identifying values. Multiple partitions of the input domain - All Combinations Coverage (ACoC), Each Choice Coverage (ECC), Pair-wise Coverage, T-wise Coverage, Base Choice Coverage, Multiple Base Choices Coverage. TriTyp example. Functional Testing - Functional Testing Concepts of Howden. Functional testing - Important Steps. Types of Functional testing -Equivalence Class Partitioning, Boundary Value Analysis, Decision Tables, Random Testing. Case Study -Black Box testing approaches using JUnit.

Module - 5 (Grey Box Testing Approaches)

Introduction to Grey Box testing - Why Grey Box testing, Gray Box Methodology, Advantages and Disadvantages. Techniques of Grey Box Testing - Matrix Testing, Regression Testing, Orthogonal Array Testing or OAT, Pattern Testing. An Introduction to PEX - Parameterized Unit Testing, The Testing Problem. Symbolic Execution – Example, Symbolic execution tree. PEX application Case Study – PEX.

TEXT BOOKS

1. Paul Ammann and JeffOffutt , Introduction to Software Testing, Cambridge University Press
2. Kshirasagar Naik and Priyadarshi Tripathy, Software Testing And Quality Assurance: Theory And Practice, Wiley.

REFERENCES

1. King, James C, “Symbolic Execution and Program Testing”, Association for Computing Machinery, July 1976.

COURSE PLAN

Module	Contents	No.of Hours
1	**Introduction to Software Testing** Some Popular Errors– Ariane 5, Therac 25, Intel Pentium Bug, What is Software testing? Why should it be tested? Software Quality, Role of Testing, Testing Process - Level 0 thinking, Level 1 thinking, Level 2 thinking, Level 3 thinking, Level 4 thinking, Software Testing Terminologies- Verification, Validation and Testing, Faults, Error and Bug, Test cases, Coverage Criteria, Types of Testing- Unit testing, integration	**9**

	testing, System testing, Acceptance testing, Beta testing, Functional testing, Stress testing, Performance testing, Usability testing and Regression testing, Testing Methods - Black Box testing, White Box testing, Grey Box testing.	
2	**Unit testing** Concept of Unit testing, Static Unit Testing, Dynamic Unit testing - Control Flow testing, Data Flow testing, Domain testing, Functional Program testing, Mutation testing - Mutation and Mutants, Mutation operators, Mutation score, Juni**t** - Framework for Unit testing, Case Study - Mutation testing using Junit, Case Study - Mutation testing using Muclipse	**9**
3	**Unit Testing:- White Box Approaches** Structural Graph Coverage Criteria - Node/vertex coverage, Edge coverage, Edge pair coverage, Path coverage, Complete path coverage, Prime path coverage, Complete round trip coverage, Simple round trip coverage, Data Flow Criteria - du paths, du pairs, Subsumption Relationships among Graph Coverage Criteria, Graph Coverage for Source Code – Control Flow Graphs (CFG) for code, CFG: If statement, CFG: If statement with return, CFG: Switch-case, CFG: Loops, CFG: Exceptions (try-catch). Example program – Statistics, Graph Coverage for Design Elements – Structural graph coverage and data flow graph coverage for design elements, Case Study - Graph Based testing using JUnit Framework. (Lecture 1), Case Study - Graph Based testing using JUnit Framework. (Lecture 2)	**9**
4	**Unit Testing:- Black Box Approaches** Domain Testing / Input Space Partitioning - Partitions of a set, Input domain modelling - Interface-based approach, Functionality-based approach, Multiple partitions of the input domain - All Combinations Coverage (ACoC), Each Choice Coverage (ECC), Pair-wise Coverage, T-wise Coverage, Base Choice Coverage, Multiple Base Choices Coverage, Functional Testing - Functional Testing Concepts of Howden. Important Steps, Types of Functional testing - Equivalence Class Partitioning, Boundary Value Analysis, Decision Tables, Random Testing, Case Study - Black Box testing approaches using JUnit	**9**
5	**Grey Box Testing Approaches** Introduction to Grey Box testing - Why Grey Box testing, Gray Box Methodology, Advantages and Disadvantages, Techniques of Grey Box Testing - Matrix Testing, Regression Testing, Orthogonal Array Testing or OAT, Pattern Testing, An Introduction to Pex - Parameterized Unit Testing, The Testing Problem, Symbolic Execution – Example, Symbolic execution tree, Case Study – PEX (Lecture 1), Case Study – PEX (Lecture 2), Case Study – PEX (Lecture 3)	**9**
		45

CS1U44F	BIOINFORMATICS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW:

This course helps the learners to understand the fundamental concepts in Molecular Biology, Genomics, Proteomics and Modelling. This course introduces bio macromolecules such as genes and proteins, different biological databases, and tools and algorithms for biological data processing, analysis and interpretation, and the elements of the systems approach to Molecular Biology. This course enables the learners to contribute towards drug discovery and computational analysis and modelling of biological process.

COURSE OUTCOMES

After the completion of the course, the student will be able to

CO #	CO	
CO 1	Describe the basic concepts of Bioinformatics with an emphasis on structure, function and synthesis of biomolecules .	Understand
CO 2	Identify biological data formats and databases, retrieve bio-sequences, and align bio-sequences to identify similarity .	Apply
CO 3	Employ similarity searching tools and algorithms to align sequences to highlight the similarity, and describe the structure of genes .	Apply
CO 4	Demonstrate Protein Structure, visualize protein structure using tools, and explain how proteins interact.	Apply
CO 5	Explain the fundamental aspects of Systems Biology, Computational Modeling and properties of models .	Understand

SYLLABUS

Module-1 (Introduction to bioinformatics)

Introduction to bioinformatics, Nature & Scope of Bioinformatics, DNA, RNA, and Protein: The Central Dogma, Messenger RNA, tRNA, rRNA, Genetic code, Gene Structure and Control, Transcription, translation

Module-2 (Introduction to bio sequences and analysis)

Introduction to Biological Databases, NCBI, Genbank, Bio sequence formats- FASTA, Sequence alignment- Global Alignment and Local Alignment, Dot Matrix Method, Dynamic Programming Method, Gap Penalties, Amino Acid Scoring Matrices - PAM and BLOSUM

Module-3 (Database Similarity Searching and genomics)

Database Similarity Searching, BLAST – Variants -BLASTN, BLASTP, BLASTX, Statistical Significance, Needleman and Wunsch and Smith–Waterman Method, Multiple Sequence Alignment, scoring function, Clustal, introduction to structure of prokaryotic and eukaryote gene

Module-4 (Proteomics)

Protein Structure, Ramachandran Plot, Hierarchies of Protein Structure, Determination of Protein three-dimensional structure, protein structure database-PDB, Protein structure visualization, introduction to Protein protein interaction, STRING database

Module-5 (Systems Biology)

Introduction to Systems Biology, Models and Modelling, Properties of models, Systems state and steady state, Variables, Parameters, and Constants in modelling, Purpose and Adequateness of Models, Advantages of Computational Modelling, Model Development, Network Versus Elements, Modularity, Robustness and Sensitivity, Data Integration

TEXT BOOKS

1. Zvelebil, Marketa J., and Jeremy O. Baum. *Understanding bioinformatics*. Garland Science, 2007.
2. Xiong, Jin. *Essential bioinformatics*. Cambridge University Press, 2006.
3. Klipp, E., Herwig, R., Kowald, A., Wierling, C., &Lehrach, H. *Systems biology in practice: concepts, implementation and application*. John Wiley & Sons. 2005

REFERENCES

1. Baxevanis, Andreas D., Gary D. Bader, and David S. Wishart, eds. *Bioinformatics*. John Wiley & Sons, 2020.
2. Shaik, Noor Ahmad, et al. *Essentials of Bioinformatics, Volume I.* Springer, 2019
3. Selzer, Paul M., Richard J. Marhöfer, and Andreas Rohwer, *Applied bioinformatics*. *An introduction–Springer, Verlag,*, 2008.
4. S C Rastogi, N Mendiratta and PRastogi, *Bioinformatics: Methods and Applications* , PHI Learning Private Limited, New Delhi, 2015.
5. D E Krane and M L Raymer, *Fundamental Concepts of Bioinformatics*, Pearson Education, 2006.
6. Andreas D.Baxevanis, B F Francis Ouellette, *Bioinformatics - A Practical Guide to the Analysis of Genes and Proteins*, Third Edition, John Wiley & Sons INC. , U.K. 2006
7. Neil C Jones and Pavel A Pevzner, *An Introduction to Bioinformatics Algorithms*, MIT press, 2004.

COURSE PLAN

Module	Contents	No. of Hours
1	**Introduction to bioinformatics** Introduction to bioinformatics, Nature & Scope of Bioinformatics, DNA, RNA, and Protein, The Central Dogma introduction, Messenger RNA, tRNA, rRNA, Genetic code, Gene Structure and Control, Transcription, Translation	**9**
2	**Introduction to bio sequences and analysis** Introduction to Biological Databases, NCBI Sequence retrieval, Genbank, Bio sequence formats- FASTA, Sequence alignment- Global Alignment and Local Alignment, Dot Matrix Method, Dynamic Programming Method, Gap Penalties, Amino Acid Scoring Matrices – PAM, BLOSUM	**9**
3	**Database Similarity Searching and genomics** Database Similarity Searching, BLAST, Variants of BLAST -BLASTN, BLASTP, BLASTX, BLAST Analysis - Statistical Significance, Needleman and Wunsch Method, Smith–Waterman Method, Multiple Sequence Alignment, scoring function, Clustal tool, Gene Structure of prokaryotic, eukaryote	**9**
4	**Proteomics** Protein Structure, Ramachandran Plot, Hierarchies of Protein Structure, Determination of Protein three-dimensional structure, protein structure database-PDB, Protein structure visualization, Protein protein interaction, Protein protein interaction networks, STRING database	**9**
5	**Systems Biology** Introduction to Systems Biology, Properties of models, Systems state and steady state, Variables, Parameters, and Constants in modelling, Purpose and Adequateness of Models, Advantages of Computational Modelling ,Model Development (introduction only), Network Versus Elements, Modularity, Robustness and Sensitivity, Data Integration	**9**
		45

CS1U44G	COMPUTATIONAL LINGUISTICS	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PEC	2	1	0	3	2020

COURSE OVERVIEW : The course aims to teach the basics of Computational Linguistics to the students viewing language phenomena from a computational/statistical standpoint. This involves ideas about statistical and computational models and how these could be linked with various language processing tasks. The course helps the learner to appreciate the complexities involved in language processing tasks using a machine, in contrast with the ease with which human beings handle them. Some practical aspects are also discussed using the Python and NLTK framework to equip the student with the capability to design solutions to linguistic problems.

COURSE OUTCOMES: After the completion of the course the student will be able to

CO1	Explain the fundamental concepts of language processing.	Understand
CO2	Demonstrate the concepts of probability, statistical inference and hidden Markov model.	Apply
CO3	Compare and summarize the various methods of word sense disambiguation, lexical acquisition and selectional preferences.	Apply
CO4	Make use of different Part-of-Speech Tagging methods for language modelling.	Apply
CO5	Examine Probabilistic Context Free Grammars and various probabilistic parsing methods	Apply
CO6	Develop simple systems for linguistic tasks using Python and NLTK.	Apply

SYLLABUS

Module- 1 (Preliminaries)

Introduction: Rationalist and Empiricist Approaches to Language-Questions that linguistics should answer-Noncategorical phenomena in language-Language and cognition as probabilistic phenomena

The Ambiguity of Language: Why natural language processing is difficult-Lexical resources-Word counts-Zipf's laws-Collocations-Concordances

Linguistic Essentials:

Parts of Speech and Morphology -Nouns and pronouns-Words that accompany nouns: Determiners and adjectives-Verbs-Other parts of speech-Phrase Structure-Phrase structure grammars -Semantics and Pragmatics-Corpus Based Work

Module -2 (Mathematical Essentials:)

Probability Theory-Probability spaces-Conditional probability and independence-Bayes' theorem-Random variables-Expectation and variance-Notation-Joint and conditional

distributions-Standard distributions-Bayesian statistics
Statistical Inference: n-gram Models over Sparse Data-Bins: Forming Equivalence Classes-Reliability vs discrimination-n gram models
Markov Models-Hidden Markov Models-Why use HMMs?-General form of an HMM-Finding the probability of an observation-Finding the best state sequence

Module -3 (Word Sense Disambiguation)
Methodological Preliminaries- Supervised and unsupervised learning-Pseudowords-Upper and lower bounds on performance-Supervised Disambiguation-Bayesian classification-Dictionary based Disambiguation-Disambiguation based on sense definitions-Thesaurus based disambiguation
Lexical Acquisition-Evaluation Measures-Verb Subcategorization -Attachment Ambiguity-PP attachment- Selectional Preferences
Semantic Similarity: Vector space measures-Probabilistic measures

Module -4 (Grammar)
Part-of-Speech Tagging-The Information Sources in Tagging-Markov Model Taggers-Hidden Markov Model Taggers-Applying HMMs to POS tagging-The effect of initialization on HMM training-Transformation Based Learning of Tags

Probabilistic Context Free Grammars-Some Features of PCFGs-Questions for PCFGs -The Probability of a String -Using inside probabilities-Using outside probabilities-Finding the most likely parse for a sentence-parsing for disambiguation-parsing model versus language model.
Module -5 (Language Processing with Python)
Introduction to NLTK, Text Wrangling and Text cleansing : Sentence Splitter, Tokenization, Stemming, Lemmatization, Stop word removal , Rare word Removal, Spell Correction. Part of Speech Tagging and NER. Parsing Structure in Text: Shallow versus deep parsing, different types of parsers and dependency parsing.

TEXT BOOKS :

1. C.D. Manning and H. Schutze. Foundations of Statistical Natural Language Processing. MIT Press.
2. Steven Bird, fEwan Klein, fEdward Loper, Natural Language Processing with Python and NLTK. O'reilly Pub.

REFERENCES

1. D. Jurafsky and J.H. Martin: Speech and Language Processing: Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, PHI.
2. James Allen: Natural Language Understanding. Pearson Pub.
3. Nitin Hardeniya, Jacob Perkins, Deepti Chopra, Nisheeth Joshi, ItiMathur: Natural Language Processing: Python and NLTK., **1**st Edition. Packt Publishing

COURSE PLAN

Module	Contents	No.of Hours
1	**Preliminaries** Introduction: Rationalist and Empiricist Approaches to Language-Questions that linguistics should answer, Non-categorical phenomena in language-Language and cognition as probabilistic phenomena, The Ambiguity of Language: Why natural language processing is difficult, Lexical resources-Word counts, Zipf's laws-Collocations-Concordances, Linguistic Essentials: Parts of Speech and Morphology -Nouns and pronouns, Words that accompany nouns: Determiners and adjectives-Verbs-Other parts of speech, Phrase Structure-Phrase structure grammars, Semantics and Pragmatics-Corpus Based Work	9
2	**Mathematical Essentials** Probability Theory-Probability spaces, Conditional probability and independence-Bayes' theorem, Random variables-Expectation and variance-Notation, Joint and conditional distributions-Standard distributions-Bayesian statistics, Statistical Inference: n-gram Models over Sparse Data-Bins: Forming Equivalence Classes, Markov Models-Hidden Markov Models: Why use HMMs?, General form of an HMM-Finding the probability of an observation-Finding the best state sequence	9
3	**Word Sense Disambiguation** Methodological Preliminaries-Supervised and unsupervised learning, Upper and lower bounds on performance-Supervised Disambiguation, Bayesian classification-Dictionary based Disambiguation, Disambiguation based on sense definitions-Thesaurus based disambiguation, Lexical Acquisition-Evaluation Measures, Verb Subcategorization-Attachment Ambiguity, PP attachment-Selectional Preferences, Semantic Similarity:Vector space measures-Probabilistic measures	9
4	**Grammar** Part-of-Speech Tagging-The Information Sources in Tagging, Markov Model Taggers-Hidden Markov Model Taggers, Applying HMMs to POS tagging-The effect of initialization on HMM training, Transformation-Based Learning of Tags, Probabilistic Context Free Grammars-Some Features of PCFGs, Questions for PCFGs, The Probability of a String -Using inside probabilities Using outside probabilities, Finding the most likely parse for a sentence-parsing for disambiguation, parsing model *vs.* language model	9
5	**Language Processing with Python** Introduction to NLTK, Text Wrangling and Text cleansing : Sentence Splitter, Tokenization, Stemming, Lemmatization, Stop word removal , Rare word Removal, Spell Correction, Part of Speech Tagging and NER, Parsing Structure in Text: Shallow versus deep parsing, types of parsers	9
		45

CS0M49B	MINI PROJECT	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PWS	0	1	6	4	2020

COURSE OVERVIEW: The objective of this course is to apply the fundamental concepts of different courses learned in respective Minor Streams: Software Engineering, Machine Learning and Networking. This course helps the learners to get an exposure to the development of application software/hardware solutions/ software simulations in the field of Computer Science and Engineering. It enables the learners to understand the different steps to be followed such as literature review and problem identification, preparation of requirement specification & design document, testing, development and deployment. Mini project enables the students to boost their skills, widen the horizon of thinking and their ability to resolve real life problems.

COURSE OUTCOMES :

After the completion of the course the student will be able to

CO#	CO	
CO1	Identify technically and economically feasible problems .	**Apply**
CO2	Identify and survey the relevant literature for getting exposed to related solutions.	**Apply**
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques .	**Apply**
CO4	Prepare technical report and deliver presentation .	**Apply**
CO5	Apply engineering and management principles to achieve the goal of the project.	**Apply**

COURSE PLAN

Students are expected to follow the following steps.

1. Review of Literature and Identification of a problem
2. Create an abstract with a problem statement, solution approach, technology stack, team, etc.
3. Create Requirements Specification
4. Create Design Document . This may include designs like,
    a. System Architecture Design
    b. Application Architecture Design
    c. GUI Design
    d. API Design

e. Database Design
f. Technology Stack

5. Deployment, Test Run & Get Results
6. Prepare Project Report

Guidelines for the Report preparation

A bonafide report on the mini project shall be submitted within one week after the final presentation. Minimum number of pages should be 40.

- Use Times New Roman font for the entire report – Chapter/Section Title – Times New Roman18, Bold; Heading 2 – Times New Roman16, Bold; Heading 3 – Times New Roman14, Bold; Body- Times New Roman 12, Normal.
- Line Spacing – Between Heading 2 – 3 lines, between lines in paragraph 1.5 lines.
- Alignments – Chapter/Section Title – Center, Heading 2 & 3 should be Left Aligned. Ensure that all body text is paragraph justified.
- Figures & Tables – Ensure that all Figures and Tables are suitably numbered and given proper names/headings. Write figure title under the figure and table title above the table.
- Suggestive order of documentation:
  i. Top Cover
  ii. Title page
  iii. Certification page
  iv. Acknowledgement
  v. Abstract
  vi. Table of Contents
  vii. List of Figures and Tables
  viii. Chapters
  ix. Appendices, if any
  x. References/Bibliography

CS1H49A	MINI PROJECT	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		PWS	0	1	6	4	2020

COURSE OVERVIEW: The objective of this course is to apply the fundamental concepts of courses learned in respective Honors Streams: Security in Computing, Machine Learning and Formal Methods. This course helps the learners to get an exposure to the development of application software/hardware solutions/ software simulations in the field of Computer Science and Engineering. It enables the learners to understand the different steps to be followed such as literature review and problem identification, preparation of requirement specification & design document, testing, development and deployment. Mini project enables the students to boost their skills, widen the horizon of thinking and their ability to resolve real life problems.

COURSE OUTCOMES : After the completion of the course the student will be able to

CO#	CO	
CO1	Identify technically and economically feasible problems **.**	**Apply**
CO2	Identify and survey the relevant literature for getting exposed to related solutions.	**Apply**
CO3	Perform requirement analysis, identify design methodologies and develop adaptable & reusable solutions of minimal complexity by using modern tools & advanced programming techniques.	**Apply**
CO4	Prepare technical report and deliver presentation **.**	**Apply**
CO5	Apply engineering and management principles to achieve the goal of the project .	**Apply**

COURSE PLAN

Students are expected to follow the following steps.

1. Review of Literature and Identification of a problem
2. Create an abstract with a problem statement, solution approach, technology stack, team, etc.
3. Create Requirements Specification
4. Create Design Document . This may include designs like,
    a. System Architecture Design
    b. Application Architecture Design

c. GUI Design
d. API Design
e. Database Design
f. Technology Stack

5. Deployment, Test Run & Get Results
6. Prepare Project Report

Guidelines for the Report preparation

A bonafide report on the mini project shall be submitted within one week after the final presentation. Minimum number of pages should be 40.

- Use Times New Roman font for the entire report – Chapter/Section Title – Times New Roman18, Bold; Heading 2 – Times New Roman16, Bold; Heading 3 – Times New Roman14, Bold; Body- Times New Roman 12, Normal.
- Line Spacing – Between Heading 2 – 3 lines, between lines in paragraph 1.5 lines.
- Alignments – Chapter/Section Title – Center, Heading 2 & 3 should be Left Aligned. Ensure that all body text is paragraph justified.
- Figures & Tables – Ensure that all Figures and Tables are suitably numbered and given proper names/headings. Write figure title under the figure and table title above the table.

- Suggestive order of documentation:
  i. Top Cover
  ii. Title page
  iii. Certification page
  iv. Acknowledgment
  v. Abstract
  vi. Table of Contents
  vii. List of Figures and Tables
  viii. Chapters
  ix. Appendices, if any
  x. References/Bibliography