Proposed Program Structure

for

B. E

(Artificial Intelligence and Machine Learning)

Scheme of Instruction and Examination

Scheme of Instruction &
Examination
B. E. - Artificial Intelligence and Machine Learning

AIML Semester - I									
S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination		Credits
			Hours Per Week			Duration in Hrs	Maximum Marks		
			L	T	P / D		CIE	SEE	
Theory Courses									
1	7BS101HS	Engineering Mathematics - 1	3	1	0	4	40	60	4
2	7BS104HS	Applied Physics	3	1	0	4	40	60	4
3	7ES101CS	Programming for Problem Solving	3	0	0	3	40	60	3
4	7ES101EE	Elements of Electrical and Electronics Engineering	3	0	0	3	40	60	3
5	7MC101CE	Environmental Science	2	0	0	2	40	60	0
Practical / Laboratory Courses									
6	7BS151HS	Applied Physics Lab	0	0	3	3	40	60	1.5
7	7ES151CS	Programming for Problem Solving Lab	0	0	2	2	40	60	1
8	7ES151EE	Elements of Electrical and Electronics Engineering Lab	0	0	2	2	40	60	1
9	7ES151CE	Engineering Graphics Lab	1	0	4	5	40	60	3
		Total Credits				**28**	**360**	**540**	**20.5**

AIML Semester - II									
S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination		Credits
			Hours Per Week			Duration in Hrs	Maximum Marks		
			L	T	P / D		CIE	SEE	
Theory Courses									
1	7BS202HS	Engineering Mathematics – II	3	1	0	4	40	60	4
2	7BS206HS	Chemistry	3	1	0	3	40	60	4
3	7HS201HS	English	2	0	0	4	40	60	2
4	7ES202CS	Data Structures	3	0	0	3	40	60	3
Practical / Laboratory Courses									
5	7BS253HS	Chemistry Lab	0	0	3	3	40	60	1.5
6	7HS251HS	English Lab	0	0	2	2	40	60	1
7	7ES252CS	Data Structures lab	0	0	2	2	40	60	1
8	7ES252ME	Engineering Workshop Practice	0	0	4	5	40	60	2
9	7MC251SP	Yoga / NSS / Sports	0	0	2	2	40	-	0
		Total Credits				**28**	**360**	**480**	**18.5**

Scheme of Instruction & Examination
B. E. - Artificial Intelligence and Machine Learning

AIML Semester - III									
S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination		Credits
			L	T	P / D	Contact Hours / week	CIE	SEE	
Theory Courses									
1	7PC301ML	Database Management Systems	4	0	0	4	40	60	4
2	7PC302ML	Computer Networks	4	0	0	4	40	60	4
3	7PC303ML	Discrete Mathematics	3	0	0	3	40	60	3
4	7ES301EC	Logic Design and Computer Architecture	4	0	0	4	40	60	4
5	7HS303HS	Human Values and Professional Ethics	3	0	0	3	40	60	2
Practical / Laboratory Courses									
6	7PC351ML	Database Management Systems Lab	0	0	2	2	40	60	1
7	7PC352ML	Python Programming Lab	0	0	2 *2	2*2	40	60	2
8	7PW353ML	Skill Development Course- I	0	0	2	2	40	60	1
		Total Credits				**26**	**320**	**480**	**21**

AIML Semester - IV									
S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination		Credits
			L	T	P / D	Contact Hours / week	CIE	SEE	
Theory -Courses									
1	7BS405HS	Probability and Statistics	3	1	0	4	40	60	4
2	7PC404ML	Operating Systems	4	0	0	4	40	60	4
3	7PC405ML	Data Warehousing and Data Mining	4	0	0	4	40	60	4
4	7PC406ML	Statistical Analytics and Computing	4	0	0	4	40	60	4
5	7MC402HS	Essence of Indian Traditional Knowledge	2	0	0	2	40	60	0
Practical / Laboratory Courses									
6	7PC454ML	Operating Systems Lab	0	0	2	2	40	60	1
7	7PC455ML	Statistical Analytics and Computing using Python Lab	0	0	2	2	40	60	1
8	7PC456ML	Data Warehousing and Data Mining Lab	0	0	2	2	40	60	1
9	7PW457ML	Skill Development Course – II	0	0	2	2	40	60	1
		Total Credits				**26**	**360**	**540**	**20**

PROFESSOR
Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University,
Hyderabad-500 007.

Head of the Department
Department of CSE
Methodist College of Engg. & Tech
Abids, Hyderabad.

Dept. of Computer Science
Methodist College of Engg. & Tech
King Koti, Hyderabad.

Scheme of Instruction & Examination
B. E. - Artificial Intelligence and Machine Learning

AIML Semester - V

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination		Credits
			L	T	P / D	Contact Hours / week	CIE	SEE	
Theory Courses									
1	7PC507ML	Design and Analysis of Algorithms	3	1	0	4	40	60	4
2	7PC508ML	Artificial Intelligence	3	1	0	4	40	60	4
3	7PC509ML	Data Science	3	1	0	4	40	60	4
4	7PE5(01 to 05) ML	Professional Elective – I	3	0	0	3	40	60	3
5	7MC503HS	Indian Constitution	2	0	0	2	40	60	0
Practical / Laboratory Courses									
6	7PC558ML	Artificial Intelligence Lab	0	0	2	2	40	60	1
7	7PC559ML	Java Programming Lab	0	0	2*2	2*2	40	60	2
8	7HS553HS	Soft Skills Lab - I	0	0	2	2	40	60	1
9	7PW560ML	Skill Development Course - III	0	0	2	2	40	60	1
		Total Credits				**27**	**360**	**540**	**20**

AIML Semester - VI

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination		Credits
			L	T	P / D	Contact Hours / week	CIE	SEE	
Theory Courses									
1	7PC610ML	Machine Learning	3	1	0	4	40	60	4
2	7PC611ML	Software Engineering	3	0	0	3	40	60	3
3	7PC612ML	Automata Languages and Compiler Design	3	1	0	4	40	60	4
4	7PE6(06 to 10) ML	Professional Elective – II	3	0	0	3	40	60	3
5	xOE601xx	Open Elective – I	3	0	0	3	40	60	3
6	7HS652HS	Effective Technical Communication	2	0	0	2	40	60	1
Practical / Laboratory Courses									
7	7PC661ML	Machine Learning Lab	0	0	2	2	40	60	1
8	7PC662ML	Software Engineering Lab	0	0	2	2	40	60	1
9	7PW663ML	Mini Project	0	0	2	2	40	60	1
		Total Credits				**25**	**360**	**540**	**21**

Professional Elective - I

1	7PE501ML	Cloud Computing
2	7PE502ML	Human Computer Interaction
3	7PE503ML	Software requirements and Estimation
4	7PE504ML	Principles of Programming Languages
5	7PE505ML	DevOps

Professional Elective - II

1	7PE606ML	Digital Forensics
2	7PE607ML	Information Retrieval Systems
3	7PE608ML	Software Project Management
4	7PE609ML	Web Technology
5	7PE610ML	Distributed Databases

Scheme of Instruction & Examination
B. E. - Artificial Intelligence and Machine Learning

AIML Semester - VII

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination		Credits
			L	T	P / D	Contact Hours / week	CIE	SEE	
Theory Courses									
1	7PC713ML	Deep Learning	3	1	0	4	40	60	4
2	7PC714ML	Generative AI	3	1	0	4	40	60	4
3	7PE7(11 to 15) ML	Professional Elective – III	3	0	0	3	40	60	3
4	7PE7(16 to 20) ML	Professional Elective – IV	3	0	0	3	40	60	3
5	xOE702xx	Open Elective - II	3	0	0	3	40	60	3
Practical / Laboratory Courses									
6	7PC764ML	Deep Learning Lab	0	0	2	2	40	60	1
7	7PW765ML	Project Work – I	0	0	4	4	40	60	2
8	7PW766ML	Summer Internship	-	-	-	-	40	60	2
		Total Credits				**23**	**360**	**540**	**22**

AIML Semester - VIII

S. No.	Course Code	Course Title	Scheme of Instruction				Scheme of Examination		Credits
			L	T	P / D	Contact Hours / week	CIE	SEE	
Theory Courses									
1	7HS802HS	Managerial Economics and Financial Accounting	3	0	0	3	40	60	3
2	7PE8(21 to 25) ML	Professional Elective – V	3	0	0	3	40	60	3
3	xOE803xx	Open Elective - III	3	0	0	3	40	60	3
Practical / Laboratory Courses									
4	7PW867ML	Project Work – II				16	50	100	8
		Total Credits				**25**	**170**	**280**	**17**

Professional Elective - III

1	7PE711ML	Ethical Hacking
2	7PE712ML	Reinforcement Learning
3	7PE713ML	Software Architecture and Design Patterns
4	7PE714ML	Scripting Languages
5	7PE715ML	Natural Language Processing

Professional Elective – IV

1	7PE716ML	Cyber Security
2	7PE717ML	Computer Vision
3	7PE718ML	Agile Methodologies
4	7PE719ML	Full Stack Development
5	7PE720ML	Blockchain Technology

Professional Elective - V

1	7PE821ML	Cryptography and Network Security
2	7PE822ML	Nature Inspired Computing
3	7PE823ML	Software Testing Methodologies
4	7PE824ML	Digital Marketing and E Commerce
5	7PE825ML	Soft Computing

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING – CREDIT STRUCTURE

Category	Sem - I	Sem - II	Sem - III	Sem - IV	Sem - V	Sem - VI	Sem - VII	Sem - VIII	Total	OU	AICTE
HS		3	2		1	1		3	11	12	12
BS	9.5	9.5		4					23	25	25
ES	11	6	4						21	26	24
PC			14	15	15	13	9		65	58	48
PE					3	3	6	3	15	18	18
OE						3	3	3	9	9	18
PW			1	1	1	1	4	8	16	13	15
MC	ES	Yoga NSS/ Sports		EITK	IC					0	Non Credit
Total	20.5	18.5	21	20	20	21	22	17	**160**	**166**	**160**
Number of Theory Subjects	5	4	5	5	5	6	5	3			
Number of Practical Subjects	4	5	3	4	4	3	3	1			

Course Summary	Inter-Departmental	Departmental	Total
Total Theory Courses	17	21	39
Total Practical Courses	08	18	26

PROFESSOR
Department of Computer Science & Engineering
University College of Engineering (A)
Osmania University,
Hyderabad-500 007.

Head of the Department
Department of CSE
Methodist College of Engg. & Tech
Abids, Hyderabad.

Dept. of Computer Science
Methodist College of Engg. & Tech
King Koti, Hyderabad.

S. No.	Networks/ Security	Data Science and Machine Intelligence	Software and Technology	Web Applications	Emerging Technologies
1	Cloud Computing	Human Computer Interaction	Software requirements and Estimation	Principles of Programming Languages	DevOps
2	Digital Forensics	Information Retrieval Systems	Software Project Management	Web Technology	Distributed Databases
3	Ethical Hacking	Reinforcement Learning	Software Architecture and Design Pattern	Scripting Languages	Natural Language Processing
4	Cyber security	Computer Vision	Agile Methodologies	Full Stack Development	Blockchain Technology
5	Cryptography and Network Security	Nature Inspired Computing	Software Testing Methodologies	Digital marketing and E Commerce	Soft Computing

V SEMESTER

DESIGN AND ANALYSIS OF ALGORITHMS

Semester V	L	T	P	Credits
Subject code – 7PC507ML	3	1	0	4

Prerequisites: Algorithms and Data Structures

Course Objectives:	Course Outcomes:
➢ Analyze the asymptotic performance of algorithms and correct proofs for algorithms ➢ Demonstrate a familiarity with major algorithms and data structures ➢ Apply important algorithmic design paradigms and methods of analysis ➢ Familiarizing students with specific algorithms for several important computational problems like sorting, searching, and graphs, etc, ➢ Introducing the concept of NP-complete problems and different techniques to deal with them	1. Understand the basic notation for analyzing the performance of the algorithms. 2. Use divide-and-conquer techniques for solving suitable problems 3. Use greedy approach to solve an appropriate problem for optimal solution. 4. Apply dynamic programming approach to solve suitable problems **5.** Understand the limitations of algorithm power and study how to cope with the limitations of algorithm power for various problems

UNIT I

Introduction & Elementary Data Structures: Introduction, Fundamentals of algorithm (Line Count, Operation Count), Analysis of algorithms (Best, Average, Worst case), Asymptotic Notations (O, Ω, Θ) Recursive Algorithms, Analysis using Recurrence Relations, Master's Theorem. Review of elementary data structures–Graphs: BFS, DFS, Articulation points, Bi-Connected Components. Sets: representation, UNION, FIND operations.

UNIT II

Divide-and-Conquer Method: The general method, Binary search, Finding maximum and minimum, Merge sort, Quick sort. Brute Force: Knapsack, Travelling salesman problem, Convex-Hull.

UNIT III

Greedy Method: Knapsack problem, Minimum spanning trees, Single source shortest path, Job sequencing with deadlines, Optimal storage on tapes, Optimal merge pattern Dynamic programming method: All pairs shortest paths, Optimal binary search tress, 0/1 Knapsack problem, Reliability design, Travelling salesman problem.

Head of the Department
Department of CSE
Methodist College of Engg. & Tech
Abids, Hyderabad.

UNIT IV

Back tracking: N-queens problem, Graph coloring, Hamiltonian cycles Branch-and-bound: FIFO & LC branch and Bound methods, 0/1 Knapsack problem, Travelling salesperson.

UNIT V

NP-Hard and NP-Complete problems: Basic concepts, non-deterministic algorithms, NP - Hard and NP-Complete classes, Cook's theorem, Proofs for NP Complete Problems: Clique, Vertex Cover.

TEXTBOOKS

1. Fundamentals of Computer Algorithms, Horowitz E, Sahni S, II Edition, Universities Press, 2007,
2. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, III Edition, PHI Learning Private Limited, 2012,

REFERENCE BOOKS

1. Algorithm Design: Foundations, Analysis and Internet Examples, Michael T. Goodrich, Roberto Tamassia, I Edition, John Wiley & Sons,2002
2. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, II Edition, Pearson education.

ARTIFICIAL INTELLIGENCE

Semester V	L	T	P	Credits
Subject code – 7PC508ML	3	1	0	4

Course Objectives:	Course Outcomes:
➢ To provide an understanding of AI fundamentals, including problem-solving, search algorithms and game-playing techniques. ➢ To teach knowledge representation methods and uncertainty handling techniques. ➢ To introduce planning techniques such as hierarchical task network, constraint satisfaction and Markov decision process. ➢ To provide an understanding of machine learning techniques for solving real-world problems. ➢ To explore AI applications in NLP, computer vision while addressing ethical concerns.	**1.** Formulate AI problems and represent them using state-space models. **2.** Understand and apply search algorithms in real-world problems. **3.** Utilize knowledge representation techniques and logical reasoning methods in problem solving and decision-making. **4.** Learn the principles of supervised and unsupervised learning, including decision trees, neural networks and reinforcement learning. **5.** Examine AI applications in NLP, computer vision, and robotics, while addressing ethical concerns.

UNIT I

Introduction to AI: Foundations of Artificial Intelligence (AI), History of AI, Structure of Agents. **Problem Solving:** Formulating problems, problem types, state space representation. **Search Strategies:** Uninformed and Informed Search (Best-first search, A* algorithm, heuristic functions). **Game Playing:** Minimax Algorithm, Alpha-Beta Pruning.

UNIT II

Knowledge-Based Systems – Basics, Knowledge Representation, Propositional Logic, First Order Logic (FOL), Inference. **Structured Knowledge Representation** – Frames, Semantic Networks. **Rule-Based & Logical Reasoning** – Deductive vs. Inductive Reasoning, Resolution. **Uncertainty Handling** – Basic Probability, Bayes' Theorem, Belief Networks, Fuzzy Logic.

UNIT III

Planning - A Simple Planning Agent, Form Problem Solving to Planning, Basic representation of plans, partial order planning. **Hierarchical Planning:** STRIPS & Hierarchical Task Networks (HTN). **Constraint Satisfaction Problems (CSPs):** AI scheduling & Constraint Solving. **Markov Decision Processes (MDP):** Decision Making in AI. **Automated Planning Applications:** Robotics, Scheduling, Smart Assistants.

UNIT IV

Learning: Machine-Learning Paradigms: Introduction, Machine Learning Systems, Supervised and Unsupervised Learning, Inductive Learning, Learning Decision Trees and Naive Bayes. **Artificial Neural Networks:** Introduction, Artificial Neural Networks, Single-Layer Feed Forward Networks, Multi-Layer Feed-Forward Networks. **Reinforcement learning:** Learning from rewards, Passive and Active reinforcement learning, Applications.

UNIT V

Communicating & Perceiving: Introduction to NLP- Progress & applications of NLP, Components of NLP, Grammars, Parsing. **Computer Vision with AI:** CNNs for Image Processing, Object Detection (YOLO, Faster R-CNN).**AI Applications:** Chatbots, Sentiment Analysis, Image Captioning, Autonomous Systems, AI in Healthcare, Finance, Robotics. **AI Ethics & Security:** Bias, Explainability, AI in Cyber security.

TEXTBOOKS

1. Artificial Intelligence – A Modern Approach, Stuart Russell and Peter Norvig, IV Edition, Pearson Education Press, 2020.
2. Artificial Intelligence, Kevin Knight, Elaine Rich, B. Nair, III Edition, McGraw Hill, 2008.
3. Artificial Intelligence, David L. Poole and Alan K. Mackworth, III Edition, Cambridge University Press, 2023.

REFERENCE BOOKS

1. Artificial Intelligence: Structures and Strategies for Complex Problem Solving, George F. Luger, VI Edition, Pearson, 2008.
2. Speech and Language Processing, Daniel Jurafsky and James H. Martin, III Edition, Pearson, 2020.

DATA SCIENCE

Semester: V	L	T	P	Credits
Subject code – 7PC509ML	3	1	0	4

Prerequisites: Statistical Analytics and Computing, Data Mining

Course Objectives:	Course Outcomes:
➢ Learn fundamental knowledge on basics of data science and R programming ➢ Learn basics of R Programming environment: R language, R- studio and R packages ➢ Understand various statistical concepts like linear and logistic regression, cluster analysis, time series forecasting ➢ Learn fundamentals of how to obtain, store, explore, and model data efficiently. ➢ Understand the concepts of classification and clustering.	1. Explain foundational concepts of data science, linear algebra, statistical inference, and R programming structures to interpret data-driven problems. 2. Develop R programs to perform arithmetic/logical operations, data frame manipulations, and predictive modeling using real-world datasets. 3. Implement clustering, classification and time series analysis in R for exploratory data analysis and pattern recognition. 4. Evaluate statistical hypotheses, confidence intervals, and p-values to infer data distributions and validate model accuracy. 5. Compare relational and NoSQ database querying techniques in R for optimal data extraction and preprocessing.

UNIT I

Data Science: Introduction to data science, Data Science process, Need for Data Science, Linear Algebra for data science, Linear equations, Distance, Eigen values, Eigen vectors

UNIT II

Descriptive statistics, data preparation. Exploratory Data Analysis, data summarization, data distribution, measuring asymmetry. Sample and estimated mean, variance and standard score. Statistical Inference frequency approach, variability of estimates, hypothesis testing using confidence intervals, using p-values.

UNIT III

Introduction to R Programming, getting started with R: Installation of R software and using the interface, Variables and data types, R Objects, Vectors and lists, Arrays, Classes, R-Programming Structures, Operations: Arithmetic, Logical and Matrix operations, Data frames, functions, Control structures, Debugging and Simulation in R

UNIT IV

Predictive Modeling: Linear Regression, Simple Linear Regression model building, Multiple Linear Regression, Logistic regression, Simulation in R.

UNIT V

Classification: performance measures, Logistic regression implementation in R, K-Nearest neighbours (KNN), K-Nearest neighbours' implementation in R, Clustering: K-Means Algorithm, K Means implementation in R. Time Series Analysis using R, Social Network Analysis, reading data from relational databases- MySQL, Reading data from NoSQL databases- MongoDB

TEXTBOOKS

1. Practical Data Science with R, Nina Zumel, II Edition, Manning Publications, 2014.
2. Practical Statistics for Data Scientists, Peter Bruce and Andrew Bruce, II- Edition, O'Reilly, 2017.
3. R for Data Science, Hadley Wickham and Garrett Grolemund, II Edition, O'Reilly, 2017

REFERENCE BOOKS

1. R Programming for Data science, Roger D Peng, Lean Publishing, 2016.
2. Introduction to Data Science, Rafael A Irizarry, Lean Publishing, 2016.
3. R Data Analysis cookbook, Vishwa Vishwanathan and Shanthi Vishwanathan 2015

CLOUD COMPUTING

Semester V	**L**	**T**	**P**	**Credits**
Subject code –7PE501ML	**3**	**0**	**0**	**3**

Prerequisites: Computer Networks

Course Objectives:	**Course Outcomes:**
➢ To provide knowledge of cloud architecture, deployment models. ➢ To introduce broad perceptive of cloud services. ➢ To introduce about storage and database management in cloud computing. ➢ To make them understand about resource management in cloud computing ➢ To make them familiar with the various cloud security issues and research trends in cloud	1. Ability to understand various service delivery models of cloud computing architecture. 2. Ability to understand the ways in which the cloud can be programmed and deployed. 3. Understand the state management database 4. Understanding cloud service providers. 5. Analyze and understand the various cloud security issues

UNIT I

Introduction - Historical Development - Cloud Computing Architecture — The Cloud Reference Model — Cloud Characteristics — Cloud Deployment Models: Public, Private, Community, Hybrid Clouds - Cloud Delivery Models: IaaS, PaaS, SaaS.

UNIT II

Cloud Computing Mechanism: Cloud Infrastructure Mechanism: Cloud Storage, Cloud Usage Monitor, Resource Replication — Specialized Cloud Mechanism: Load Balancer, SLA Monitor, Hypervisor, Resource Cluster, Multi Device Broker

UNIT III

State Management Database — Cloud Management Mechanism: Remote Administration System, Resource Management System, SLA Management System, State Management Database

UNIT IV

Security in the Cloud: Basic Concepts - Threat Agents - Cloud Security Threats - Cloud Security Mechanism: Encryption, Hashing, Digital Signature, Public Key Infrastructure. Identity and Access Management. Data Security: Application Security-Virtual Machine Security.

UNIT V

Case Studies: Google App Engine (GAE) — GAE Architecture — Functional Modules o GAE — Amazon Web Services (AWS) — GAE Applications — Cloud Software Environments Eucalyptus — Open Nebula — Open Stack.

TEXTBOOKS

1. Cloud Computing, Concept, Technology and Architecture, Thomas Erl, Zaigham Mahmood, Ricardo Puttini, Prentice Hall, 2013.
2. Cloud Computing, A Practical Approach, Toby Velte, Anthony Velte, Robert C. Elsenpeter, Tata McGraw-Hill Edition, 2010.
3. Cloud Computing: Implementation, Management, And Security, Rittinghouse, John W., and James F. Ransome, CRC Press, 2017.

REFERENCE BOOKS

1. Cloud Computing: Principles and Paradigms, Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, Wiley, 2011.
2. Distributed and Cloud Computing, Kai Hwang, Geoffery C. Fox, Jack J. Dongarra, Elsevier, 2012.
3. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Kumaraswamy, Shahed Latif, O'Reilly, SPD, 2011

HUMAN COMPUTER INTERACTION

Semester V	**L**	**T**	**P**	**Credits**
Subject code: 7PE502ML	**3**	**0**	**0**	**3**

Prerequisites: Software Engineering, Web Technologies

Course Objectives:	**Course Outcomes:**
➢ To gain an overview of Human-Computer Interaction (HCI), ➢ To understand user interface design and alternatives to traditional "keyboard and mouse" computing ➢ To become familiar with the vocabulary associated with sensory and cognitive systems as relevant to task performance by humans ➢ To apply models from cognitive psychology to predicting user performance in various human-computer interaction tasks ➢ To analyze the importance of a design and evaluation methodology that begins with and maintains a focus on the user.	1. Explain the capabilities of both humans and computers from the viewpoint of human information processing. 2. Describe typical human–computer interaction (HCI) models, styles, and various historic HCI paradigms. 3. Apply an interactive design process and universal design principles to designing HCI systems. 4. Describe and use HCI design principles, standards and guidelines. 5. Analyze and identify user models, user support, socio-organizational issues, and stakeholder requirements of HCI systems.

UNIT I

Introduction: Importance of user Interface – definition, importance of good design. Benefits of good design. A brief history of Screen design.

The graphical user interface – popularity of graphics, the concept of direct manipulation, graphical system, Characteristics, Web user – Interface popularity, characteristics- Principles of user interface.

UNIT II

Design process – Human interaction with computers, importance of human characteristics human consideration, Human interaction speeds, understanding business junctions.

Screen Designing: Design goals – Screen planning and purpose, organizing screen elements, ordering of screen data and content – screen navigation and flow – Visually pleasing composition – amount of information – focus and emphasis – presentation information simply and meaningfully – information retrieval on web – statistical graphics – Technological consideration in interface design.

UNIT III

Windows – New and Navigation schemes selection of window, selection of devices based and screen-based controls. Components – text and messages, Icons and increases – Multimedia, colours, uses problems, choosing colours

UNIT IV

HCI in the software process: The software life cycle Usability Engineering Iterative design and prototyping Design Focus: Prototyping in practice Design rationale Design rules Principles to support usability Standards Golden rules and heuristics HCI patterns Evaluation techniques, Goals of evaluation, Evaluation through expert analysis, Evaluation through user participation, Choosing an evaluation method. Universal design, Universal design principles Multi-modal interaction

UNIT V

Cognitive models Goal and task hierarchies Design Focus: GOMS saves money, Linguistic models, The challenge of display-based systems, Physical and device models, Cognitive architectures, Ubiquitous computing and augmented realities, Ubiquitous computing applications research, Design Focus: Ambient Wood – augmenting the physical, Virtual and augmented reality, Design Focus: Shared experience, Design Focus: Applications of augmented reality Information and data visualization

TEXTBOOKS

1. The essential guide to user interface design, Wilbert O Galitz, Wiley Dream Tech, 2002.
2. Human – Computer Interaction. Alan Dix, Janet Fincay, GreGoryd, Abowd, Russell Bealg, Pearson Education, 2003.

REFERENCE BOOKS

1. Designing the user interface. III Edition Ben Shneidermann, Pearson Education Asia, 2009.
2. Interaction Design Prece, Rogers, Sharps, V Edition, Wiley Dreamtech, 2019.
3. User Interface Design, Soren Lauesen, Addison-Wesley, 2004.
4. Human –Computer Interaction, D. R. Olsen, Cengage Learning, 2009.

SOFTWARE REQUIREMENTS AND ESTIMATION

Semester: V
Subject code: 7PE503ML
Prerequisites: Software Engineering

L	T	P	Credits
3	0	0	3

Course Objectives:	Course Outcomes
➢ To introduce good practices for requirements engineering ➢ To understand requirements elicitation and elicitation techniques ➢ To learn the usage of analysis models and software quality attributes ➢ To acquire knowledge on software estimation, size estimation, effort, schedule and cost estimation ➢ Perform software project reviews and evaluation according to best practices	1. Gain knowledge about software requirements, requirements management, their principles and practices 2. Analyze requirement elicitation techniques and prototyping 3. Gain knowledge about requirement management, their principles and practices 4. Analyze use-case modelling and different data diagrams 5. Estimate software in terms of size, cost, effort and schedule

UNIT I

Software Requirements: Essential Software Requirements, Good practices for requirements engineering, Improving requirements processes, Software requirements and risk management. **Software Requirements Engineering:** Requirements elicitation, requirements analysis documentation, review, elicitation techniques, analysis models, Software quality attributes, risk reduction through prototyping, setting requirements priorities, verifying requirements quality.

UNIT II

Software Requirements Management: Requirements management Principles and practices, Requirements attributes, Change Management Process, Requirements Traceability Matrix, Links in requirements chain

Software Requirements Modeling: Use Case Modeling, Analysis Models, Data flow diagram, state transition diagram, class diagrams, Object analysis, Problem Frames

UNIT III

Software Estimation: Components of Software Estimations, Estimation methods, Problems associated with estimation, Key project factors that influence estimation.

Size Estimation: Two views of sizing, Function Point Analysis, Mark IIFPA, Full Function Points, LOC Estimation, Conversion between size measures.

UNIT IV

Effort, Schedule and Cost Estimation: Productivity, Estimation Factors, Approaches to Effort and Schedule Estimation, COCOMO II, Putnam Estimation Model, Algorithmic models, Cost Estimation

UNIT V

Tools for Requirements Management and Estimation Requirements Management Tools: Benefits of using a requirements management tool, commercial requirements management tool, Rational Requisite pro, Caliber – RM, implementing requirements management automation.

Software Estimation Tools: Desirable features in software estimation tools, IFPUG, USC's COCOMO II, SLIM (Software Life Cycle Management) Tools.

TEXTBOOKS

1. Software Requirements and Estimation, Swapna Kishore, Rajesh Naik, I Edition, Tata Mc GrawHill,2001
2. Software Requirements, Karl E. Weigers, Edition II, Microsoft Press,2003

REFERENCE BOOKS

1. Managing Software Requirements, Dean Leffingwell& Don Widrig, Pearson Education, 2003.
2. Mastering the requirements process, Suzanne Robertson & James Robertson, II Edition, Pearson Education, 2006.
3. Estimating Software Costs, Capers Jones, II Edition, TMH, 2007.

PRINCIPLES OF PROGRAMMING LANGUAGES

Semester: V

Subject code: 7PE504ML

L	T	P	Credits
3	0	0	3

Prerequisites: C, Data Structures and Algorithms, Discrete Mathematics

Course Objectives	Course Outcomes:
➢ To understand the fundamental concepts of principles of language design, formal syntax and semantic, BNF. ➢ To understand different data types, variables, expressions, types of statements, different types of control statements and iterations. ➢ To understand the concept of Sub programs and blocks, operator overloading, and co-routines. ➢ To understand the concept of Abstract data types, concurrency, exception handling of different programming languages and logic programming languages ➢ To understand Functional Programming Languages like FPL, LISP, ML languages.	1. Ability to express syntax and semantics in formal notation. 2. Ability to apply suitable programming paradigm for the application. 3. Gain Knowledge and comparison of the features of programming languages 4. Program in different language paradigms and evaluate their relative benefits. 5. identify and describe semantic issues associated with variable binding, scoping rules, parameter passing, and exception handling.

UNIT I

Preliminary Concepts: Reasons for studying, concepts of programming languages, Programming domains, Language Evaluation Criteria, influences on Language design, Language categories, Programming Paradigms – Imperative, Object Oriented, functional Programming, Logic Programming. Programming Language Implementation – Compilation and Virtual Machines, programming environments. Syntax and Semantics: general Problem of describing Syntax and Semantics, formal methods of describing syntax - BNF, EBNF for common programming languages features, parse trees, ambiguous grammars, attribute grammars, denotation semantics and axiomatic semantics for common programming language features.

UNIT II

Data types: Introduction, primitive, character, user defined, array, associative, record, union, pointer and reference types, design and implementation use related to these types. Names, Variable, concept of binding, type checking, type compatibility, named constants, variable initialization. Expressions and Statements: Arithmetic relational and Boolean expressions, short circuit evaluation mixed mode assignment, Assignment Statements, Control Structures – Statement Level, Compound Statements, Selection, Iteration, Unconditional Statements, guarded commands

UNIT III

Software Estimation: Components of Software Estimations, Estimationmethods, Problems
Subprograms Blocks and Fundamentals of sub-programs: Scope and lifetime of variable, static and dynamic scope, Design issues of subprograms, local referencing environments, parameter passing methods, overloaded sub-programs, generic sub-programs, parameters that are subprogram names, design issues for functions user defined overloaded operators, co routines.

UNIT IV

Abstract Data Types: Abstractions and Encapsulation, Introduction to Data Abstraction, Design Issues, Object Oriented Programming in C++, Java,Ada 95.
Exception Handling: Exceptions, Exception Propagation, Exception Handler in Ada, C++ and Java.
Logic Programming Language: Introduction and Overview of Logic Programming, Basic Elements of Prolog, Application of Logic Programming.

UNIT V

Functional Programming Languages: Introduction, fundamentals of FPL, LISP, ML application of Functional Programming Languages and comparison of functional and imperative Languages. Key Concepts, Case Study: Python – Values and Types, Variables, Storage and Control, Bindings and Scope, Procedural Abstraction, Data Abstraction, Separate Compilation, Module Library.

TEXTBOOKS

1. Concepts of Programming Languages Robert. W. Sebesta 8/e, Pearson Education, 2008.
2. Programming Languages Design Concepts, D. A. Watt, Wiley Dreamtech, rp – 2007

REFERENCE BOOKS

1. Programming languages, 2nd Edition A. B. Tucker, R E Noonan, TMH
2. Programming Languages, K C Louden, 2nd Edition, Thomson, 2003.
1. LISP Patric Henry Winston and Paul Horn Pearson Education.
2. Programming in PROLOG W F Clocksin& C S Mellish, V Edition, Springer

DEVOPS

Semester V	L	T	P	Credits
Subject code: 7PE505ML	**3**	**0**	**0**	**3**

Prerequisites: Basics of Programming languages, Cloud Computing,

Course Objectives:	Course Outcomes
➢ Describe the agile relationship between development and IT operations ➢ Understand the skill sets and high-functioning teams involved in DevOps and related methods ➢ Implement automated system update and DevOps lifecycle. ➢ Understand concepts of DevOps maturity and assessment.	1. Identify components of DevOps environment. 2. Describe Software development models and architectures of DevOps. 3. Apply different project management, integration, testing and code deployment tools. 4. Investigate different DevOps Software development models. 5. Assess various DevOps practices

UNIT I

Introduction: Introduction, Agile development model, DevOps, and ITIL. DevOps process and Continuous Delivery, Release management, Scrum, Kanban, delivery pipeline, bottlenecks, examples.

UNIT II

Software development models and DevOps: DevOps Lifecycle for Business Agility, DevOps, and Continuous Testing. **DevOps influence on Architecture:** Introducing software architecture, The monolithic scenario, Architecture rules of thumb, The separation of concerns, Handling database migrations, Micro services, and the data tier, DevOps, architecture, and resilience.

UNIT III

Introduction to project management: The need for source code control, The history of source code management, Roles and code, source code management system and migrations, Shared authentication, Hosted Git servers, Different Git server implementations, Docker intermission, Gerrit, The pull request model, GitLab.

UNIT IV

Integrating the system: Build systems, Jenkins build server, managing build dependencies, Jenkins plugins, and file system layout, The host server, Build slaves, Software on the host, Triggers, Job chaining and build pipelines, Build servers and infrastructure as code, building by dependency order, Build phases, Alternative build servers, Collating quality measures.

UNIT V

Testing Tools and automation: Various types of testing, Automation of testing Pros and cons, Selenium - Introduction, Selenium features, JavaScript testing, testing backend integration points, Test-driven development, REPL-driven development

Deployment of the system: Deployment systems, Virtualization stacks, code execution at the client, Puppet master and agents, Ansible, Deployment tools: Chef, Salt Stack and Docker

TEXTBOOKS

1. Practical Devops, Joakim Verona. II Edition, Packt Publishing, 2018.
2. DevOps Tools from Practitioner's Viewpoint, Deepak Gaikwad, Viral Thakkar, Wiley publications, 2020.

REFERENCE BOOKS

1. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison Wesley; ISBN-10.
2. Effective DevOps, Jennifer Davis & Ryn Daniels, IV Edition, O'Reilly
3. The DevOps Handbook, Gene Kim, Jez Humble, Patrick Debois, John Willis, First Edition, IT Revoultion Press, LLC

ARTIFICIAL INTELLIGENCE LAB

Semester V	L	T	P	Credits
Subject code – 7PC558ML	**0**	**0**	**2**	**1**

Prerequisites: Python Programming

Course Objectives:	Course Outcomes:
➢ Equip students with practical knowledge of search algorithms for AI problem-solving. ➢ Develop skills in decision-making strategies through game-playing and heuristic approaches. ➢ Provide hands-on experience in supervised learning techniques for classification tasks. ➢ Introducing fundamental text processing techniques for Natural Language Processing ➢ Encourage students to explore real-world AI applications through project-based learning.	1. Apply search algorithms to solve AI problems. 2. Design and Implement decision making algorithms. 3. Develop and evaluate supervised machine learning models. 4. Implement text processing and natural language techniques. 5. Develop a case study to demonstrate the use of AI.

List of Programs

1. Develop a program to solve the Water Jug Problem using Breadth First Search technique.
2. Write a program to solve 5-queens problem using Depth First Search technique.
3. Design and implement solutions for 8-puzzle problems using A* search algorithm.
4. Develop a python program to solve maze navigation problems using greedy best first search.
5. Implement Mini max algorithm for finding an optimal decision in a tic-tac toe game.
6. Implement an AI agent for Hangman Game.
7. Develop a program to train, evaluate and compare the performance of the following classifiers using scikit-learn on a dataset:
   a. Decision Tree
   b. Multi-layer Feed Forward neural network
8. Implement Text processing using NLTK to
   a. Remove stop words
   b. Implement stemming
   c. POS (Parts of Speech) tagging

In addition to the above programs, students should be encouraged to study and implement one of the following

- Game bot (Tic Tac toe, 7 puzzle, Snake Game AI)
- Text classification
- Chat bot

JAVA PROGRAMMING LAB

Semester V	L	T	P	Credits
Subject code – 7PC559ML	0	0	2*2	2

Prerequisites: PPS

Course Objectives:	Course Outcomes:
➢ To implement various java concepts. ➢ To write java programs to solve mathematics, science and engineering problems. ➢ To identify compile time and runtime errors, syntax and logical errors ➢ To import the essentials of java class library and user defined packages. ➢ To develop skills in internet programming using applets and swings.	1. To understand the use of OOPs concepts. 2. Develop Java program using packages, inheritance and interface. 3. Develop java programs to implement error handling techniques using exception handling. 4. Develop graphical user interface using AWT. 5. Demonstrate event handling mechanism.

List of Programs

1. Implement the concept of classes and objects.
2. Implement Arrays to a given application.
3. Use String and String Tokenizer classes and develop java programs.
4. Develop a java program Using interfaces and packages.
5. Develop Java Programs using inheritance.
6. Develop Java programs using Method overloading and method overriding.
7. Develop java programs using Exception handling (using try, catch, throw, throws and
8. finally).
9. Develop java programs using Multithreading (using Thread class and Runnable interface, synchronization).
10. Develop java programs using collections (using list, set, Map and generics).
11. CASE STUDY: Develop a program to calculate SGPA & CGPA of a student and display the progress report.

INPUT:

INPUT		
ROLLNO	NAME	HOWMANYSEMESTERS? Semester wise: Subject Code, Subject Name And Marks

OUTPUT:

Progress report of <NAME>

Roll No:

Program (BE/ME):

Branch:

College Code and Name:

Year of joining:

Semester-I Grades	**Semester-II Grades**	**Semester-III Grades**
Subject 1:	Subject 1:	Subject 1:
Subject 2:	Subject 2:	Subject 2:
Subject 3:	Subject 3:	Subject 3:
....	....	....
SGPA:	SGPA:	SGPA:
CGPA:	CGPA:	CGPA:

Note: The above experiments can be implemented using any IDE.

REFERENCE BOOKS

1. JAVA, The complete reference, Thirteenth Edition, Herbert Schildt, Danny Coward

SKILL DEVELOPMENT COURSE- III

Semester V	L	T	P	Credits
Subject code –7PW560ML	0	0	2	1

Prerequisites:

Course Objectives:	Course Outcomes:
➢ Able to identify the basic components of an Android app, such as activities, layouts, and views. ➢ Be able to use layouts to arrange your user interface elements in a logical and efficient way. ➢ Be able to store data in the app's internal storage, or in a cloud-based storage service. ➢ Able to add that feature to an existing Android app. ➢ Able to deploy that app to the Google Play Store.	1. Understand the basics of Android development, including the Android Studio IDE, the Android SDK, and the AndroidManifest.xml file. 2. Create an app with multiple activities that can communicate with each other using intents. 3. Create a variety of user interface elements, such as buttons, text fields, and checkboxes. 4. Use layouts to arrange their user interface elements in a logical and efficient way. **5.** 5.Understand how to store data in Android apps, using both local and remote storage options.

List of Programs:

1. **Portable Devices Overview**

1.1. Introduction to SW development for portable devices

1.2. Overview of Portable Devices

1.3. HW & SW for Portable Devices

1.4. Applications of Portable Devices

1.5. Portable devices - Understanding HW platforms

1.5.1. HW Platforms (Processors, Peripheral devices, Sensors etc)

1.5.2. HW Platforms – Mobile Phones + Wireless

1.5.3. HW Platforms – Internet of things (IoT) + Wireless

1.5.4. Example - Rasberry Pi

1.5.5. Sensors in Portable devices

1.5.6. Generic HW platforms

2. Overview of SW Platforms & Development

2.1. Mobile OS

2.1.1. Architecture and Framework of different mobile platforms

2.1.2. Development platforms and development tools

2.1.3. Programming languages

2.1.4. Simulator and emulator

2.1.5. SDK and Development Environments

2.1.6. Development Life Cycle of Application

2.2. CREATING APPLICATIONS AND ACTIVITIES

2.2.1. Introducing the Application Manifest File

2.2.2. Creating Applications and Activities

2.2.3. Architecture Patterns (MVC)

2.2.4. Review of other Architecture and Design patterns

2.2.5. The Android Application Lifecycle

3. User Interface Design; Intents and Broadcasts

3.1. Fundamental Android UI Design

3.2. Introducing Layouts

3.3. Introducing Fragments

3.4. Introducing Intents

3.5. Creating Intent Filters and Broadcast Receivers

4. Background Services and Using Internet Resources

4.1. Introducing Services

4.2. Using Background Threads

4.3. Parsing Internet Resources

4.4. Using the Download Manager

4.5. Using Internet Services

4.6. Connecting to Google App Engine

4.7. Best Practices for Downloading Data Without Draining the Battery

5. Files, Saving States and Preferences

5.1. Shared Preferences

5.2. Introducing the Preference Framework and the Preference Activity

5.3. Static Files as Resources

5.4. Working with the File System

6. Database and Content Providers

6.1. Introducing Android Databases

6.2. Introducing SQLite

6.3. Content Values and Cursors

6.4. Working with SQLite Databases

6.5. Creating Content Providers

6.6. Using Content Providers

6.7. Case Study: Native Android Content Providers

7. Location Based Services, Telephony and SMS

7.1. Using Location-Based Services

7.2. Using the Emulator with Location-Based Services
7.3. Selecting a Location Provider
7.4. Using Proximity Alerts
7.5. Using the Geocoder
7.6. Example: Map-based activity
7.7. Hardware Support for Telephony
7.8. Using Telephony
7.9. Introducing SMS and MMS

8. **Hardware Support and Devices (AUDIO, VIDEO, AND USING THE CAMERA)**

8.1. Using Sensors and the Sensor Manager
8.2. Monitoring a Device's Movement and Orientation
8.3. Introducing the Environmental Sensors
8.4. Playing Audio and Video
8.5. Using Audio Effects
8.6. Using the Camera
8.7. Recording Video
8.8. Adding Media to the Media Store

TEXTBOOKS

1. Professional Android 4 Application Development, by Reto Meier, WROX Press, Wiley Publishing

REFERENCE BOOKS

1. Android Application Development, Programming with the Google SDK, by, Rick Rogers, John Lombardo, Zigurd Mednieks, Blake Meike, SPD, Oreilly, ISBN10: 81-8404-733-9, ISBN13:978-81-8404-733-2

2. Hello Android, Introducing Google's Mobile Development Platform, 3rd Edition, by Ed Burnette, Pragmatic Programmers, LLC.ISBN-10: 1-934356-56-5, ISBN-13: 978-1-934356-56-2

VI SEMESTER

MACHINE LEARNING

Semester VI	L	T	P	Credits
Subject code –7PC610ML	3	1	0	4

Prerequisites: Mathematics, Python, Data Structures and Algorithms

Course Objectives:	Course Outcomes:
➢ To introduce the core concepts, applications, and types of Machine Learning. ➢ To explore Supervised Learning techniques and evaluate their performance. ➢ To understand and implement Unsupervised Learning techniques. ➢ To explore Instance-Based and Analytical Learning approaches. ➢ To implement Reinforcement Learning and optimization strategies in ML systems.	1. Understand the basic concepts, applications, and types of Machine Learning 2. Apply and analyze Supervised Learning models and evaluate performance 3. Implement Unsupervised Learning techniques 4. Compare and apply Instance-Based and Analytical Learning 5. Demonstrate knowledge of RL and Optimization Strategies

UNIT I

Introduction to Machine Learning: Applications of Machine Learning, Representation and Learning: Feature vectors, Feature space, Types of Machine Learning - Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning, Parametric and Non-Parametric Machine Learning Algorithms.

UNIT II

Supervised Learning: Regression: Linear Regression, Multiple Linear Regression, Evaluation Measures: MAE, SSE, RMSE, R2.
Classification: Logistic Regression, Decision Trees, Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), Naive Bayes Classifier, Performance Metrics – Accuracy, Precision, Recall, F1-score, and Confusion Matrix, Overfitting and Underfitting, Bias-Variance Tradeoff, ROC Curve, AUC.

UNIT III

Ensemble Algorithms: Bagging, Boosting, Random Forest.
Unsupervised Learning: Clustering: k-Means Clustering, Hierarchical Clustering, DBSCAN, Fuzzy C-means algorithm, Evaluation Metrics – Silhouette Score.
Dimensionality Reduction and Feature Selection: Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Autoencoder

UNIT IV

Instance-based learning: Introduction, K-Nearest Neighbor learning, Locally weighted Regression, Radial Basis Functions.
Analytical Learning: Introduction, Learning with Perfect Domain Theories: PROLOG-EBG, Remarks on Explanation-Based Learning, Explanation-Based Learning of Search Control Knowledge.

UNIT V

Reinforcement Learning and Optimization: Markov Decision Process, Q-Learning, Deep Q Networks (DQN), Policy Gradient Methods, Genetic Algorithms, Hyperparameter Tuning - Grid Search, Random Search, Bayesian Optimization.

TEXTBOOKS

1. Tom M. Mitchell, "Machine Learning", McGraw-Hill, 1st Edition.
2. Christopher M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2nd Edition.
3. Aurélien Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow", O'Reilly Media, 2nd Edition

REFERENCE BOOKS

1. Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 1st Edition.
2. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep Learning", MIT Press, 1st Edition.

SOFTWARE ENGINEERING

Semester VI	L	T	P	Credits
Subject code – 7PC611ML	3	0	0	3

Prerequisites: Oops Concepts.

Course Objectives:	Course Outcomes:
➢ Knowledge of the process structure, layered technologies, engineering techniques and procedures, waterfall and evolutionary software process models. ➢ Understanding the basic methods and procedures of SW engineering and SRS needs data models, object models, context models, and behavioral models. ➢ To provide an awareness of the practical knowledge of the methods for design, Knowledge of various software architecture styles and its estimation, ➢ To expertise with software testing techniques, such as unit and integration testing and gaining Knowledge of methods for validation and verification. ➢ Discussing software testing methods for validation and verification, including risk management, evaluations, and product metrics, including static analysis	1. Ability to translate end-user requirements into system and software requirements, using e.g. 2. Ability to design UML patterns, and structure the requirements in a Software Requirements Document (SRD). 3. Identify and apply appropriate software architectures and patterns to carry out high level projects 4. Design of a system and be able to critically compare alternative choices. 5. Evaluate and experience testing problems and will be able to develop a simple testing report.

UNIT I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths.

A Generic view of process: Software engineering- a layered technology, a process framework. the capability maturity model integration (CMMI), process patterns, process assessment. personal and team process models.

Process models: The waterfall model, incremental process models, evolutionary process models, the unified process.

UNITII

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document.

Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

System models: Context models, behavioural models, data models, object models, structured methods.

UNIT III

Design Engineering: Design process and design quality, design concepts, the design model.
Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design,
Conceptual model of UML: Basic structural modelling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.
Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

UNIT V

Metrics for Process and Products: Software measurement, metrics for software quality.
Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

TEXTBOOKS

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, Mc Graw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

REFERENCE BOOKS

1. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The Mc Graw-Hill Companies.
3. Fundamentals of object-oriented design using UML Meiler page-Jones: Pearson Education.

AUTOMATA LANGUAGES AND COMPILER DESIGN

Semester VI	L	T	P	Credits
Subject code – 7PC612ML	3	1	0	4

Prerequisites: Discrete Mathematics, Compiler Design

Course Objectives:	Course Outcomes:
➢ To introduce the concept of formal specification of languages and different classes of formal languages ➢ To Discuss automata models corresponding to different levels of Chomsky hierarchy ➢ To Analyze and explain the behavior of push-down automata and TM. ➢ To teach concepts of language translation and phases of compiler design ➢ To inculcate knowledge of parser by parsing LL parser and LR parser	1. Explain finite state machines for modeling and their power to recognize the languages. 2. Summarize the concept of Regular languages and context free languages. 3. Construct PDA and Turing machines for the given set of languages. 4. Build the lexical and Syntax analyser phases of compiler. 5. 5.Model SDD's using Intermediate Representations

UNIT I

Formal Language and Regular Expressions: Languages, Definition Languages regular expressions, Finite Automata – DFA, NFA. Conversion of regular expression to NFA, NFA to DFA. Finite automata with output – Moore and Mealy machines, conversion of Moore to Mealy and Mealy to Moore

UNIT II

Context Free grammars and parsing: Context free grammars, derivation, parse trees, ambiguity, Leftmost and rightmost derivation of strings and Sentential forms, Ambiguity, left recursion and left factoring in context free grammars, Minimization of context free grammars, Normal forms for context free grammars, Chomsky normal form

Pushdown Automata: Introduction to Pushdown automata, Acceptance of context free languages, Acceptance by final state and acceptance by empty state and its equivalence, Equivalence of context free grammars and pushdown automata.

UNIT III

Turing Machine: Introduction to Turing Machine, Design of Turing machines, Types of Turing machines.

Introduction to Compiling: Overview of Compilers, Phases of a Compiler.

Lexical Analysis: The Role of Lexical Analyzer, Input Buffering, Specification of Tokens, Recognition of Tokens, LEX tool.

UNIT IV

Syntax Analysis: The role of the Parser, First and Follow, Predictive Parsing
Bottom up parsing: Shift reduce parsing, LR Parsers-SLR, Canonical LR, LALR, Parser Generator (YACC).
Semantic Analysis: Syntax directed translation, S-attributed and L-attributed grammars
Intermediate code generation – abstract syntax tree, Three address code, Implementations

UNIT V

Run time storage: Storage organization, storage allocation strategies
Code optimization: Optimization of basic blocks, peephole optimization, flow graphs, Data flow analysis of flow graphs.
Code generation: Machine dependent code generation, Register allocation and assignment. Using DAG representation of Block.

TEXTBOOKS

1. Introduction to Automata Theory Languages and Computation, John E Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, III Edition, Pearson Education, 2011.
2. Compilers- Principles Techniques and Tool, Alfred Aho, Monica S Lam, RaviSethi, Jeffrey D.Ullman, II Edition, Pearson Education India, 2013.

REFERENCE BOOKS

1. An introduction to Formal Languages and Automata, Peter Linz, VI Edition, Jones & Bartlett, 2016
2. Principles of Compiler Design, V. Raghavan, I Edition, McGrawHill Education, 2017.
3. Theory of Computer Science – Automata Languages and Computation, Mishra and Chandrashekaran, III Edition, PHI, 2009
4. Formal Languages and Automata Theory, K.V.N.Sunitha , N.Kalyani, I Edition, TMH, 2010.
5. Introduction to Theory of Computation, Michel Sipser, II Edition, Thomson, 2012.

DIGITAL FORENSICS

Semester: VI	**L**	**T**	**P**	**Credits**
Subject Code: 7PE606ML	**3**	**0**	**0**	**3**

Prerequisites: Computer Networks, Database management Systems

Course Objectives:	**Course Outcomes:**
➢ To understand the basic digital forensics and techniques for conducting forensic examination on different digital devices. ➢ To understand how to examine digital evidences such as data acquisition, identification analysis. ➢ To Understand the processing crimes and incident scenes ➢ To Understand the latest computer forensic tools.	1. Apply forensic analysis tools to recover important evidence for identifying computer crime. 2. Understand computing investigation 3. Understand the perspective of data acquisition tools 4. Understand the process of digital crimes 5. Understand the latest computer forensic tools.

UNIT I

Computer forensics fundamentals, Benefits of forensics, computer crimes, computer forensics evidence and courts, legal concerns and private issues.

UNIT II

Understanding Computing Investigations – Procedure for corporate High-Tech investigations, understanding data recovery workstation and software, conducting and investigations.

UNIT III

Data acquisition- understanding storage formats and digital evidence, determining the best acquisition method, acquisition tools, validating data acquisitions, performing RAID data acquisitions, remote network acquisition tools, other forensics acquisitions tools.

UNIT IV

Processing crimes and incident scenes, securing a computer incident or crime, seizing digital evidence at scene, storing digital evidence, obtaining digital hash, reviewing cases.

UNIT V

Current computer forensics tools- software, hardware tools, validating and testing forensic software, addressing data-hiding techniques, performing remote acquisitions, E-Mail investigations- investigating email crime and violations, understanding E-Mail servers, specialized E-Mail forensics tool.

TEXTBOOKS

1. Computer Forensics: Incident Response Essentials, Warren G. Kruse II and Jay G. Heiser, Addison Wesley, 2002.
2. Guide to Computer Forensics and Investigations, Nelson, B, Phillips, A, Enfinger, F, Stuart, C., II Edition, Thomson Course Technology, 2006.

REFERENCE BOOKS

1. Computer Forensics, Computer Crime Scene Investigation, Vacca, J, II Edition, Charles River Media, 2005.
2. Digital Forensics Explained, Greg Gogolin, II Edition, CRC Press, 2021.
3. Cybersecurity and Digital Forensics, Challenges and Future Trends, Mangesh M Ghonge, Sabyasachi Pramanik, I Edition, Scrivener Publishing, 2022.

INFORMATION RETRIVAL SYSTEM

Semester: VI	L	T	P	Credits
Subject Code: 7PE607ML	3	0	0	3

Prerequisites:

Course Objectives:	Course Outcomes
➢ To learn the different models for information storage and retrieval ➢ Learning about the various retrieval utilities ➢ To understand indexing and querying in information retrieval systems ➢ To expose the students to the notions of structured and semi structured data ➢ To learn about web searches	1. Understands to store and retrieve textual documents using appropriate models 2. Uses the various retrieval utilities for improving search 3. Understands the indexing and compressing documents to improve space and time efficiency 4. Formulates SQL like queries for unstructured data 5. Understand issues in web search

UNIT I

Introduction, Retrieval Strategies: Vector space model, Probabilistic retrieval strategies: Simple term weights, Non binary independence model, Language Models.

UNIT II

Retrieval Utilities: Relevance feedback, Clustering, N-grams, Regression analysis, Thesauri.

UNIT III

Retrieval Utilities: Semantic networks, Parsing

Cross-Language Information Retrieval: Introduction, Crossing the language barrier.

UNIT IV

Efficiency: Inverted index, Query processing, Signature files, Duplicate document detection.

UNIT V

Integrating Structured Data and Text: A Historical progression, Information retrieval as a relational application, Semi-structured search using a relational schema
Distributed Information Retrieval: A Theoretical model of distributed retrieval, Web search.

TEXTBOOKS

1. Information Retrieval – Algorithms and Heuristics, Springer, David A. Grossman, Ophir

Frieder, II Edition (Distributed by Universities Press)
2. Information Storage and Retrieval Systems, Gerald J Kowalski, Mark T Maybury, II Edition, Springer, 2000

REFERENCE BOOKS

1. Mining the Web: Discovering Knowledge from Hypertext Data, Soumen Chakrabarti, Morgan-Kaufmann Publishers, 2002
2. An Introduction to Information Retrieval, Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Cambridge University Press, Cambridge, England, 2009
3. Modern Information Retrieval by Yates and Neto, IV Edition, Pearson Education.

SOFTWARE PROJECT MANAGEMENT

	L	T	P	Credits
Semester: VI				
Subject Code: 7PE608ML	**3**	**0**	**0**	**3**

Prerequisites:

Course Objectives:	Course Outcomes:
➢ To understand software project planning and evaluation techniques ➢ To plan and manage projects at each stage of the software development life cycle(SDLC) ➢ To learn about the activity planning and risk management principles ➢ To acquire skills to manage various phases involved in project management and people management	1. Understand the basic project management concepts, framework and the process models 2. Apply appropriate software process model and software effort estimation techniques 3. Estimate risks involved in various project activities, staff and issues related to people management 4. Analyze checkpoints, project reporting structure, project progress and tracking mechanisms using project management principles

UNIT I

PROJECT EVALUATION AND PROJECT PLANNING

Importance of Software Project Management – Activities – Methodologies –Categorization of Software Projects – Setting objectives – Management Principles –Management Control – Project portfolio Management – Cost-benefit evaluation technology – Risk evaluation – Strategic program Management – Stepwise Project Planning

UNIT II

PROJECT LIFE CYCLE AND EFFORT ESTIMATION

Software process and Process Models – Choice of Process models – Rapid Application development – Agile methods – Dynamic System Development Method – Extreme Programming– Managing interactive processes – Basics of Software estimation – Effortand Cost estimation techniques – COSMIC Full function points – COCOMO II – a Parametric Productivity Model

UNIT III

ACTIVITY PLANNING AND RISK MANAGEMENT

Objectives of Activity planning – Project schedules – Activities – Sequencing and scheduling – Network Planning models – Formulating Network Model – Forward Pass & Backward Pass techniques – Critical path (CRM) method – Risk identification –Assessment – Risk Planning –Risk Management – – PERT technique – Monte Carlo simulation – Resource Allocation – Creation of critical paths – Cost schedules.

UNIT IV

PROJECT MANAGEMENT AND CONTROL

Framework for Management and control – Collection of data – Visualizing progress –Cost monitoring – Earned Value Analysis – Prioritizing Monitoring – Project tracking –Change control – Software Configuration Management – Managing contracts – Contract Management

UNIT V

STAFFING IN SOFTWARE PROJECTS

Managing people – Organizational behavior – Best methods of staff selection – Motivation– The Oldham – Hackman job characteristic model – Stress – Health and Safety – Ethicaland Professional concerns – Working in teams – Decision making – Organizational structures – Dispersed and Virtual teams – Communications genres – Communication plans – Leadership

TEXTBOOKS

1. Software Project Management, Bob Hughes, Mike Cotterell and Rajib Mall, V Edition Tata McGraw Hill, New Delhi, 2012
2. Effective Software Project Management, Robert K. Wysocki, Wiley Publication, 2011

REFERENCE BOOKS

1. Software Project Management, Walker Royce, Addison-Wesley, 1998
2. Managing Global Software Projects, Gopalaswamy Ramesh, McGraw Hill EducatioN (India), Fourteenth Reprint 2013

WEB TECHNOLOGY

Semester: VI	L	T	P	Credits
Subject Code: 7PE609ML	3	0	0	3

Prerequisites:

Course Objectives:	Course Outcomes:
➢ To introduce PHP language for server-side scripting ➢ To introduce HTML for the design of pages. ➢ To introduce XML and processing of XML Data with Java ➢ To introduce Server-side programming with Java Servlets and JSP ➢ To introduce Client-side scripting with JavaScript and AJAX.	1. Gain knowledge of client-side scripting, validation of forms and AJAX programming 2. Understand server-side scripting with PHP language 3. Understand what is XML and how to parse and use XML Data with Java 4. To introduce Server-side programming with Java Servlets and JSP 5. Develops appropriate client side scripting programs using Java Script and AJAX(Applying)

UNIT- I

Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, Reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads. Connecting to database (My SQL as reference), executing simple queries, handling results, Handling sessions and cookies
File Handling in PHP: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories.

UNIT- II

HTML Common tags- List, Tables, images, forms, Frames; Cascading Style sheets;
XML: Introduction to XML, Defining XML tags, their attributes and values, Document Type Definition, XML Schemes, Document Object Model, XHTML Parsing XML Data – DOM and SAX Parsers in java.

UNIT - III

Introduction to Servlets: Common Gateway Interface (CGt), Life cycle of a Servlet, deploying a servlet, The Servlet API, Reading Servlet parameters, Reading Initialization parameters, Handling Http Request & Responses, Using Cookies and Sessions, connecting to a database using JDBC.

UNIT - IV

Introduction to JSP: The Anatomy of a JSP Page, JSP Processing, Declarations,

Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages, Using Cookies and session for session tracking, connecting to database in JSP.

UNIT - V

Client-side Scripting: Introduction to Javascript, Javascript language – declaring variables, scope of variables, functions. Event handlers (onclick, onsubmitetc.), Document Object Model, Form validation.

TEXT BOOKS

1. Web Technologies, UttamK Roy, Oxford UniversityPress
2. The Complete Reference PHP — Steven Holzner, TataMcGraw-Hill

REFERENCE BOOKS

1. Web Programming, building internet applications, Chris Bates 2" edition, WileyDreamtech
2. Java Server Pages —Hans Bergsten, SPDO'Reilly,
3. Java Script,D.Flanagan
4. Beginning Web Programming-Jon DuckettWROX.
5. Programming world wide web, R.W.Sebesta, Fourth Edition,Pearson.
6. Internet and World Wide Web — How to program. Dietel and Nieto,Pearson.

DISTRIBUTED DATABASES

	L	T	P	Credits
Semester: VI				
Subject Code: 7PE610ML	**4**	**0**	**0**	**4**

Prerequisites:

Course Objectives:	Course Outcomes:
➢ To learn Principles and Levels of Distribution Transparency. ➢ To study Distributed Database Design, Query Processing and its Optimization. ➢ To learn Management of Distributed Transactions and Concurrency Control. ➢ To learn Reliability and Heterogeneous Distributed Database Administration. ➢ To learn use of R* Project Database.	1. Understand the Principles of Distributed Databases and types of Data Fragmentation with levels of Distribution Transparency. 2. Design Distributed Database and analyze the background processes involved in queries, assess and apply database query optimization. 3. Understand issues surrounding management and concurrency control in Distributed Database. 4. Explain Reliability, Catalog Management and Problems of Heterogeneous Distributed Database. 5. Create and use new database data types and Inheritance. Apply logic in R* Project database.

UNIT I

Principles of Distributed Database: Introduction to Distributed Database, Distributed Vs Centralized Database.

Levels of Distribution Transparency: Reference Architecture for Distributed Database. Types of Data Fragmentation, Distributed Transparency, Integrity Constraints in Distributed Database, Distributed Database Access Primitives.

UNIT II

Distributed Database Design: Framework for Distributed Database, Design of Distributed Database, Database Fragmentation, Allocation of Fragments.

Overview of Query Processing: Translation of Global Queries into Fragment Queries, Query Optimization, Join Processing, General Queries, Parametric Queries, Distributed Grouping and Aggregate Function, Evaluation.

UNIT III

Management of Distributed Transactions: A Framework of Transaction Management, Supporting Atomicity of Distributed Transaction, Concurrency Control for Distributed Transaction, Architectural Aspects of Distributed Transactions.

Concurrency Control: Foundation of Distributed Concurrency Control, Serializability in Distributed Database, 2-Phase Locking, Timestamps, Distributed Deadlocks

UNIT IV

Reliability: Basic Concepts, Reliability and Concurrency Control, Determining a Consistent View of the Network.
Distributed Database Administration: Catalog Management in Distributed Databases, Authorization and Protection.
Heterogeneous Distributed Database Systems: Problems of Heterogeneous Distributed Database Systems, **DDTS:** A Distributed Testbed System

UNIT V

The R* Project: Architecture of R*, Protocols for Data Definition and Authorization in R*.
Distributed Object Database Management Systems: Complex Data Types, Structured Types and Inheritance in SQL, Table Inheritance, Object Identity and Reference Types in SQL, Object Oriented Vs Object Relational.

TEXTBOOKS

1. Distributed Databases: Principles and Systems, Stefano Ceri, Giuseppe Palgatti, Third Edition, McGraw Hill.
2. Database Management System, Majumdar and Bhattacharya, Second Edition, TMH.
3. Database System Concepts, Henry F. Korth, Abraham Silberschatz, Third Edition, McGraw Hill.

REFERENCE BOOKS

1. Fundamentals of Database Systems, Ramez Elmasri, Shamkanth Navathe, Sixth Edition, Addison Wesley.
2. Principles of Distributed Database Systems, M. Tamer Ozsu, Patrick Valduriez, Second Edition, Springer.

MACHINE LEARNING LAB

Semester – VI	L	T	P	Credits
Subject code -7PC661ML	0	0	2	1

Prerequisites: Python Programming

Course Objectives:	Course Outcomes:
➢ To Learn to install and work with Python, Anaconda, and essential ML libraries. ➢ To apply regression, classification, clustering, and ensemble techniques. ➢ To Implement clustering and dimensionality reduction techniques. ➢ To perform hyperparameter tuning and apply Genetic Algorithms for optimization. ➢ To apply learned concepts to case studies and practical datasets.	➢ To set up a machine learning environment and preprocess datasets using Python libraries like NumPy, Pandas, and Scipy. ➢ To implement and compare classification algorithms such as Logistic Regression. Decision Trees, K-NN, Naïve Bayes, and SVM. ➢ To apply clustering techniques like k-Means and Hierarchical Clustering and interpret the obtained clusters. ➢ To perform dimensionality reduction using PCA, LDA, and optimize models using Genetic Algorithms. ➢ To evaluate ML models using performance metrics and apply them in real-world case studies.

List of Programs

1. Basic Data Preprocessing
   a. Installation of Python environment/Anaconda IDE for machine learning: installing Python modules/Packages like scikit-learn, Keras, and Tensorflow etc.
   b. Programs involving pandas, Numpy, and Scipy libraries.
2. Programs for classification
   a. Build models using linear regression and logistic regression and apply it to classify a new instance
3. Write a program to demonstrate the following classifiers. Use an appropriate data set for building the model. Apply the model to classify a new instance.
   a. Decision tree
   b. K nearest neighbor (K-NN)
   c. Naïve Bayes
   d. Support Vector Machine
4. Demonstration of Clustering algorithms using
   a. k-means
   b. Hierarchical algorithms (agglomerative etc). Interpret the clusters obtained.
5. Demonstrate ensemble techniques like boosting, bagging, random forests etc.
6. Build a classifier, and compare its performance with an ensemble technique like random forest.
7. Evaluate various classification algorithms performance on a dataset using various measures like True Positive rate, False positive rate, precision, recall etc.
8. Dimensionality Reduction & Feature Selection

a. Implement Principal Component Analysis (PCA) on a high-dimensional dataset.
b. Apply Linear Discriminant Analysis (LDA) for feature selection.

9. Demonstrate GA for optimization (minimization or maximization problem).
10. Case study on supervised/unsupervised learning algorithm

SOFTWARE ENGINEERING LAB

Semester – VI	L	T	P	Credits
Subject code – 7PC662ML	**3**	**0**	**0**	**3**

Prerequisites: Programming for Problem Solving

Co-requisite

1. A Course on "Software Engineering"

Course Objectives:	Course Outcomes:
➢ To have hands on experience in developing a software project by using various software techniques. ➢ Ability to generate Engineering principles and methods in each of the phases of software development. ➢ To implement various software designs, data flow diagram models. ➢ Apply data models, object models, context models and behavioural models.	1. Ability to translate end-user requirements into system and software requirements 2. Ability to generate a high-level design of the system from the software requirements 3. Ability to Understand and develop various structure and behaviour UML diagrams. 4. Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

List of Experiments

Do the following 8 exercises for any two projects given in the list of sample projects or any other projects:

1. Development of problem statement.
2. Preparation of Software Requirement Specification Document, Design Documents and
2. Testing Phase related documents.
3. Preparation of Software Configuration Management and Risk Management related
4. documents.
5. Study and usage of any Design phase CASE tool
6. Performing the Design by using any Design phase CASE tools.
7. Develop test cases for unit testing and integration testing
8. Develop testcases for various white box and black box testing techniques.

Sample Projects:

1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system

6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
9. E-book management System.
10. Recruitment system

TEXTBOOKS

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, Mc Graw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

MINI PROJECT

Semester – VI	L	T	P	Credits
Subject code – 7PW663ML	0	0	2	1

Prerequisites:

Course Objectives:	Course Outcomes:
➢ To enhance practical and professional skills. ➢ To familiarize tools and techniques of systematic literature survey and documentation ➢ To expose students to industry practices and teamwork ➢ To encourage students to work with innovative and entrepreneurial data	1. Demonstrate the ability to synthesize and apply knowledge and skills acquired in the academic program to real world problems 2. Evaluate different solutions based on economic and technical feasibility 3. Effectively plan a project and confidently perform all aspects of project management 4. 4.Develop and test the solution

Guidelines for Mini Project

1. The mini project is a team activity with a maximum of 3 students in a team. This is software-based design work.
2. The mini project may be a combination of hardware and software
3. Mini Project should cater to a small system required in laboratory or real life.
4. After interactions with course coordinator and based on comprehensive literature survey/ need analysis, the student shall identify the title and define the aim and objectives of mini project.
5. Students are expected to detail specifications, methodology, resources required, critical issues involved in design and implementation and submit the proposal within the first week of the semester.
6. The student is expected to exert on design, development and testing of the proposed work as per the schedule.
7. Completed mini project and documentation in the form of mini project report is to be submitted at the end of semester.