

**LORDS INSTITUTE OF ENGINEERING AND TECHNOLOGY**  
**(An Autonomous Institution)**  
**DEPARTMENT OF INFORMATION TECHNOLOGY**  
**SCHEME OF INSTRUCTIONS & EXAMINATIONS [LR-21]**  
**(W.E.F Academic Year 2023-24)**  
**B.E. V-Semester**

S. No.	Course Code	Course Category	Course Title	Scheme of Instructions				Scheme of Examination			CREDITS
				L	T	P/D	Contact Hours/Week	CIE	SEE	Duration in Hours	
<b>Theory Course</b>											
1	U21MB501	HSMC	Business Economics And Financial Analysis	3	-		3	40	60	3	3
2	U21CM501	PCC	Automata Theory, Languages And Computation	3	-	-	3	40	60	3	3
3	U21CS501	PCC	Design & Analysis of Algorithms	3	1	-	4	40	60	3	4
4	U21IT501	PCC	Software Engineering	3	-	-	3	40	60	3	3
5	--	OEC	Open Elective – I	3	-	-	3	40	60	3	3
<b>Practical / Laboratory Course</b>											
6	U21IT5L1	PCC	Software Engineering Lab	-	-	3	3	25	50	3	1.5
7	U21IT5L2	PCC	Scripting Languages Lab	-	-	2	2	50	-	-	1
<b>Project</b>											
8	U21IT5P1	PROJ	Internship (During vacation period after 4Sem.)	-	-	2	2	50	-	-	1
<b>Skill Development Course</b>											
9	U21MA5L1	BSC	Aptitude and Reasoning	-	-	2	2	25	50	3	1
<b>Total</b>				<b>15</b>	<b>1</b>	<b>9</b>	<b>25</b>	<b>350</b>	<b>400</b>	<b>-</b>	<b>20.5</b>

**L:** Lecture (Hrs/Wk/Sem) **T:** Tutorial (Hrs/Wk/Sem) **P:** Practical / **D:** Drawing (Hrs/Wk/Sem)  
**CIE:** Continuous Internal Evaluation **SEE:** Semester End Examination (Univ.Exam)  
**OEC:** Open Elective Courses **PCC:** Programme Core Courses  
**BSC:** Basic Science Courses **MA:** Mathematics  
**HSMC:** Humanities and Social Sciences **IT:** Information Technology  
**PCC:** Programme Core Courses **PROJ:** Project  
**MB:** Master of Business Administration

**Note:**

- Each contact hour is a Clock Hour.
- The duration of the practical class is two hours, however it can be extended wherever necessary, to enable the student to complete the experiment.

**LORDS INSTITUTE OF ENGINEERING AND TECHNOLOGY**  
**(An Autonomous Institution)**  
**DEPARTMENT OF INFORMATION TECHNOLOGY**  
**SCHEME OF INSTRUCTIONS & EXAMINATIONS [LR-21]**  
**(W.E.F Academic Year 2023-24)**  
**B.E. VI-Semester**

S. No.	Course Code	Course Category	Course Title	Scheme of Instructions				Scheme of Examination			CREDITS
				L	T	P/D	Contact Hours/Week	CIE	SEE	Duration in Hours	
<b>Theory Course</b>											
1	U21IT601	PCC	Artificial Intelligence	3	-	-	3	40	60	3	3
2	U21IT602	PCC	Computer Networks	3	-	-	3	40	60	3	3
3	U21CD601	PCC	Machine Learning	3	-	-	3	40	60	3	3
4	--	PEC	Professional Elective – I	3	-	-	3	40	60	3	3
5	--	OEC	Open Elective – II	3	-	-	3	40	60	3	3
<b>Practical / Laboratory Course</b>											
6	U21IT6L1	PCC	Artificial Intelligence Lab	-	-	3	3	25	50	3	1.5
7	U21IT6L2	PCC	Computer Networks Lab	-	-	3	3	25	50	3	1.5
8	U21CD6L2	PCC	Machine Learning Lab	-	-	3	3	25	50	3	1.5
<b>Project</b>											
9	U21IT6P1	PROJ	Mini Project	-	-	6	6	50	50	3	3
<b>Total</b>				<b>15</b>	<b>-</b>	<b>15</b>	<b>30</b>	<b>325</b>	<b>500</b>	<b>-</b>	<b>22.5</b>

**L:** Lecture (Hrs/Wk/Sem) **T:** Tutorial (Hrs/Wk/Sem) **P:** Practical / **D:** Drawing (Hrs/Wk/Sem)  
**CIE:** Continuous Internal Evaluation **SEE:** Semester End Examination (Univ.Exam)  
**OEC:** Open Elective Courses **HSMC:** Humanities and Social Sciences  
**IT:** Information Technology **PCC:** Programme Core Courses  
**PE:** Professional Elective **EN:** English  
**PROJ:** Project

**Note:**

- Each contact hour is a Clock Hour.
- The duration of the practical class is three hours, however it can be extended wherever necessary, to enable the student to complete the experiment.

**LORDS INSTITUTE OF ENGINEERING AND TECHNOLOGY**  
**(An Autonomous Institution)**  
**DEPARTMENT OF INFORMATION TECHNOLOGY**  
**SCHEME OF INSTRUCTIONS & EXAMINATIONS**  
**(W.E.F Academic Year 2021-22)**  
**PROFESSIONAL ELECTIVE COURSES**

S. No.	Course Code	Category	Course Title	Scheme of Instructions				Scheme of Examination			CREDITS
				L	T	P/D	Contact Hours/Week	Maximum Marks		Duration in Hours	
								CIE	SEE		
<b>Theory Course</b>											
1	U21IT603	PEC-I	Principles of Programming Languages	3	0	0	3	40	60	3	3
	U21IT604		Data Mining	3	0	0	3	40	60	3	3
	U21IT605		Advanced Operating Systems	3	0	0	3	40	60	3	3
	U21IT606		Software Testing Methodologies	3	0	0	3	40	60	3	3
2	U21IT703	PEC-II	Distributed Systems	3	0	0	3	40	60	3	3
	U21IT704		Data Science with R Programming	3	0	0	3	40	60	3	3
	U21IT705		Information Security	3	0	0	3	40	60	3	3
	U21IT706		Software Testing and Quality Assurance	3	0	0	3	40	60	3	3
3	U21IT707	PEC-III	Cloud Computing	3	0	0	3	40	60	3	3
	U21IT708		Deep Learning	3	0	0	3	40	60	3	3
	U21IT709		Ethical Hacking	3	0	0	3	40	60	3	3
	U21IT710		Software Reuse Techniques	3	0	0	3	40	60	3	3
4	U21IT801	PEC-IV	Scalable architectures for Large Applications	3	0	0	3	40	60	3	3
	U21IT802		Natural Language Processing	3	0	0	3	40	60	3	3
	U21IT803		Web Security	3	0	0	3	40	60	3	3
	U21IT804		Software Project Management	3	0	0	3	40	60	3	3
5	U21IT805	PEC-V	Fog Computing	3	0	0	3	40	60	3	3
	U21IT806		Computational Intelligence	3	0	0	3	40	60	3	3
	U21IT807		Digital Forensics	3	0	0	3	40	60	3	3
	U21IT808		Agile Software Development	3	0	0	3	40	60	3	3

**PROFESSIONAL ELECTIVE COURSES WITH 4 THREADS**

<b>6<sup>th</sup> Sem</b>	<b>7<sup>th</sup> Sem</b>		<b>8<sup>th</sup> Sem</b>	
<b>PE-I</b>	<b>PE-II</b>	<b>PE-III</b>	<b>PE-IV</b>	<b>PE-V</b>
Principles of Programming Languages	Distributed Systems	Cloud Computing	Scalable architectures for Large Applications	Fog Computing
Data Mining	Data Science with R Programming	Deep Learning	Natural Language Processing	Computational Intelligence
Advanced Operating Systems	Information Security	Ethical Hacking	Web Security	Digital Forensics
Software Testing Methodologies	Software Project Management	Software Quality Assurance	Agile Software Development	Software Reuse Techniques

**LORDS INSTITUTE OF ENGINEERING AND TECHNOLOGY**  
**(An Autonomous Institution)**  
**DEPARTMENT OF INFORMATION TECHNOLOGY**  
**SCHEME OF INSTRUCTIONS & EXAMINATIONS**  
**(W.E.F Academic Year 2021-22)**  
**OPEN ELECTIVE COURSES**

S. No.	Course Code	Category	Course Title
1	U21EE508	OEC 1	Non Conventional Energy Systems
	U21EE509		Energy Conservation and Management
	U21CS508		Data Base Management Systems
	U21IT502		Data Structures
	U21CD603		Data Ethics
	U21ME509		Basics of Mechanical Engineering
	U21ME510		Modern Manufacturing Processes
	U21CE509		Disaster Preparedness and Management
	U21CE510		Green Building Technologies.
	U21EC507		Principles of Communication Theory
	U21EC508		Basic Electronics
	U21MB501		Managerial Communication
	U21MB503		Managerial Science and Theory
	U21SH501		History of Science & Technology
	U21SH502		Economic Policies in India

S. No.	Course Code	Category	Course Title
2	U21EE608	OEC 2	Fundamental of Power Electronics
	U21EE609		Electrical Installation and Safety
	U21CS607		Java Programming
	U21IT607		Introduction to Web Programming
	U21ME608		Basics Of 3-D Printing
	U21ME609		Optimization Methods for Engineers
	U21CE608		Construction Materials
	U21CE609		Road Safety Engineering
	U21EC607		Principles of Data Communication and Network
	U21EC608		Embedded Systems
	U21MB602		Total Quality Management
	U21MB603		Innovation Management
	U21SH601		Indian Music System
	U21SH602		Introduction to Art and Aesthetics

Note: Open elective subjects not offered to the students of own/CSE & allied department

**LORDS INSTITUTE OF ENGINEERING AND TECHNOLOGY**  
**(An Autonomous Institution)**  
**DEPARTMENT OF INFORMATION TECHNOLOGY**  
**SCHEME OF INSTRUCTIONS & EXAMINATIONS**  
**(W.E.F Academic Year 2021-22)**  
**OPEN ELECTIVE COURSES OFFERED**

S. No.	Course Code	Category	Course Title
3	U21EE711	OEC 3	Introduction to Electrical Vehicles
	U21EE712		Design estimation and Costing of Electrical Systems
	U21CS711		Data Sciences
	U21IT711		Basics of Artificial Intelligence
	U21ME711		Renewable Energy Resources
	U21ME712		Cooling of Electronic Components
	U21CE711		Environmental Systems
	U21CE712		Urban Transportation System
	U21EC703		IOT and its protocols
	U21EC704		Television and Video Engineering
	U21MB702		Logistics Management
	U21MB703		Management of Start Up's
	U21SH701		Display Devices
	U21SH702		Comparative Study of Literature

S. No.	Course Code	Category	Course Title
4	U21EE804	OEC 4	Smart Building Systems
	U21EE805		Industrial Automation
	U21CS806		Basics of Machine Learning
	U21IT809		Introduction to Cloud Computing
	U21ME806		Hybrid Vehicle Technology
	U21ME807		Power Plant Engineering
	U21CE806		Green Building Technology
	U21CE807		Environmental Impact Assessment
	U21EC805		Fundamentals of Wireless Communication
	U21EC806		Fundamental Digital Design using Verilog HDL
	U21MB802		Entrepreneurship
	U21MB803		E - Marketing
	U21SH801		Corrosion Science and Technology
	U21SH802		Introduction To Philosophical Thoughts

Course Code	Course Title				Core/Elective		
U21MB501	BUSINESS ECONOMICS AND FINANCIAL ANALYSIS				CORE		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
	3	-	-	-	40	60	3

**Course Objectives**

Develop ability to

1. Understand the concepts of Business and Economics
2. Comprehend the concepts of market dynamics namely elasticity of demand and pricing in different market structures.
3. Gain the knowledge on the production theories and cost analysis while dealing with the production and the concept of breakeven analysis.
4. Acquire the in depth knowledge on Financial Accounting concepts and principles and preparation of final accounts.
5. Understand the financial statements through ration analysis and cash flow techniques.

**Course Outcomes**

At the end of the course, student would be able to

1. Apply the concepts of business and economics during his professional and personal life.
2. Understand the elasticity of the demand of the product, different types, and measurement of elasticity of demand and factors influencing on elasticity of demand.
3. Recognize the Production function, features of Iso-Quants and Iso-Costs, different types of internal economies, external economies and law of returns with appropriate examples.
4. Prepare the financial statements of the firm.
5. Analyze the financial statements using ratio analysis and cash flow techniques.

**UNIT – I**

**Introduction to Business and Economics: Business:** Structure of Business Firm, Theory of Firm, Types of Business Entities, Sources of Capital for a Company, Non-Conventional Sources of Finance. **Economics:** Significance of Economics, Micro and Macro Economic Concepts, Business Cycle, Features and Phases of Business Cycle. Nature and Scope of Business Economics, Multidisciplinary nature of Business Economics.

**UNIT – II**

**Demand and Supply Analysis:** Elasticity of Demand: Elasticity, Types of Elasticity, Law of Demand, Measurement and Significance of Elasticity of Demand, Factors affecting Elasticity of Demand, Elasticity of Demand in decision making, Demand Forecasting: Characteristics of Good Demand Forecasting, Steps in Demand Forecasting, Methods of Demand Forecasting.

**UNIT- III**

**Production, Cost, and Market Structures & Pricing: Production Analysis:** Factors of Production, Production Function, Production Function. **Cost analysis:** Types of Costs, Short run and Long run Cost Functions. **Market Structures:** Features of Perfect competition, Monopoly, Oligopoly, and Monopolistic Competition. **Pricing:** Types of Pricing, Product Life Cycle based Pricing, Break Even Analysis, & Cost Volume Profit Analysis.

**UNIT – IV**

**Financial Accounting:** Accounting concepts and Conventions, Accounting Equation, Double Entry system of Accounting, Rules for maintaining Books of Accounts, Journal, Posting to Ledger, Preparation of Trial Balance, Elements of Financial Statements, and Preparation of Final Accounts.

## **UNIT - V**

**Financial Analysis through Ratios:** Concept of Ratio Analysis, Liquidity Ratios, Turnover Ratios, Profitability Ratios, Proprietary Ratios, Solvency, Leverage Ratios (simple problems).

### **Suggested Readings:**

1. A.R. Aryasri, "Managerial Economics and Financial Analysis", TMH Publications, 3rd Edition, 2007.
2. D.N. Dwivedi, "Managerial Economics", Vikas Publication House Pvt. Ltd, 2nd Edition, 2012.
3. S.N. Maheshwari & S.K.Maheshwari, "Financial Accounting", Vikas Publication House Pvt.Ltd, 4th Edition, 2012.
4. R. Narayana Swamy, "Financial Accounting- A managerial Perspective", Pearson publications, 1st Indian Reprint Edition, 2012.
5. J.V. Prabhakar Rao & P.V. Rao, "Managerial Economics & Financial Analysis", Maruthi Publishers, 1st Revised Edition, 2011.
6. M.Kasi Reddy & Saraswathi, "Managerial Economics and Financial Analysis", PHI Publications, New Delhi, 10th Revised Edition, 2012.
7. Varshney & Maheswari, "Managerial Economics", Sulthan Chand Publishers, 1st Revised Edition, 2009.

Course Code	Course Title				Core/Elective		
U21CM501	AUTOMATA THEORY, LANGUAGES AND COMPUTATION				CORE		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Discrete Mathematics	L	T	D	P			
	3	-	-	-	40	60	3
<b>Course Objectives</b> Develop ability to <ol style="list-style-type: none"> <li>1. Develop a formal notation for strings, languages and machines.</li> <li>2. Understand Regular Expression and algebraic laws.</li> <li>3. Design context free grammars and PDA, to generate strings from a context free language and Convert them into normal forms.</li> <li>4. Identify the hierarchy of formal languages, grammars and machines.</li> <li>5. Distinguish between computability and non-computability and Decidability and undecidability.</li> </ol> <b>Course Outcomes</b> At the end of the course, student would be able to <ol style="list-style-type: none"> <li>1. Gain knowledge of the various abstract machines.</li> <li>2. Use regular languages and regular expression for constructing different finite state machines.</li> <li>3. Understand and design different types of grammars.</li> <li>4. Construct Push down Automata.</li> <li>5. Construct Turing Machine.</li> </ol>							

**UNIT-I**

**Introduction:** Introduction to Finite Automata, Structural Representations, Basic Concepts of Automata Theory – Symbol, Alphabets, Strings, Languages, Kleene Closure, Positive Closure.

**Finite Automata without Output:** Deterministic Finite Automata, Nondeterministic Finite Automata, Finite Automata with Epsilon-Transitions, Conversion of NFA to DFA

**UNIT-II**

**Regular Expressions:** Finite Automata and Regular Expressions, Applications of Regular Expressions, Algebraic Laws for Regular Expressions, Properties of regular sets, Conversion of Finite Automata to Regular Expressions.

**Pumping Lemma for Regular Languages,** Statement of the pumping lemma, Applications of the Pumping Lemma.

**Closure Properties of Regular Languages:** Closure properties of Regular languages, Decision Properties of Regular Languages, Minimization of Automata- Equivalence, Myhill - Nerode Theorem

**UNIT-III**

**Context-Free Grammars:** Definition of Context-Free Grammars, Derivations Using a Grammar, Leftmost and Rightmost Derivations, the Language of a Grammar, Sentential Forms, Parse Trees, Ambiguity in Grammars.

**Push down Automata:** Definition of the Pushdown Automaton, the Languages of a PDA, Equivalence of PDA's and CFG's, Acceptance by final state & empty stack, Deterministic PDA Conversion of CFG to PDA, Conversion of PDA to CFG

**UNIT-IV**

**Normal Forms for Context- Free Grammars:** Grammar Simplification, Chomsky Normal form, Greibach Normal form.

**Context-Free Languages:** Statement of pumping lemma, Applications pumping lemma

**Properties of Context-Free Languages:** Closure properties of CFL's, Decision Properties of CFL's

### UNIT-V

**Turing Machines:** Introduction, Techniques for construction of Turing machines, Modifications of TM, TM as enumerator, Restricted TM.

**Undecidability:** Recursive and Recursively enumerable languages, Halting problem, Rice Theorem, Post's correspondence problem, P, NP, NP-Complete, NP-Hard Problems, Chomsky's Hierarchy–Regular grammars, Unrestricted grammar, CSL

### Suggested Reading:

1. Introduction to Automata Theory, languages, and Computation, John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, Pearson Education India, 3rd Edition, 2008.
2. Automata and Computability, Undergraduate Texts in Computer Science, Dexter C. Kozen, Springer, 2007.
3. Introduction to the Theory of Computation, Michael Sipser, PWS Publishing 3rd Edition, 2014.
4. Introduction to Languages and the Theory of Computation, John Martin, Tata McGraw Hill, 3rd Edition, 2002.

Course Code	Course Title				Core/Elective		
U21CS501	DESIGN AND ANALYSIS OF ALGORITHMS				Core		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
Data Structures	3	1	-	-	40	60	4
<p><b>Course Objectives:</b> The objectives of this course are:</p> <ol style="list-style-type: none"> <li>1. Acquire the knowledge of Algorithm and problem-solving technique, limitations of algorithms.</li> <li>2. Understand different techniques like divide and conquer, transfer and conquer etc., to solve problems.</li> <li>3. Understand different techniques like greedy method and dynamic programming</li> <li>4. Understand the concepts of Np-Hard And Np-Complete.</li> <li>5. Illustrating the methods of backtracking and branch bound techniques to solve the problems like N-queens problem, graph coloring</li> </ol> <p><b>Course Outcomes</b> At the end of the course, the students will be able to:</p> <ol style="list-style-type: none"> <li>1. Identify asymptotic notations and basic efficiency classes.</li> <li>2. Solve problems using various techniques like divide-and-conquer and transfer-and-conquer.</li> <li>3. Use different algorithms like TSP, Floyd's etc. to solve real world problems.</li> <li>4. Introduce the P and NP classes.</li> <li>5. Develop solutions for n - Queens problem, Subset – Sum Problem, Assignment problem, Knapsack problem etc.</li> </ol>							

**UNIT - I**

**Introduction:** Algorithm, Fundamentals of algorithmic problem solving, Fundamentals of the analysis of algorithm efficiency, Asymptotic Notations and basic efficiency classes, Mathematical Analysis of Non-Recursive and Recursive Algorithms, The substitute method, Recursion tree method, Master method.

**UNIT - II**

**Divide and conquer:** Divide and Conquer: General Method, Binary Search, finding minimum and maximum MergeSort analysis, Quick Sort analysis, Strassen's matrix multiplication.

**Transfer and conquer:** Introduction, Balanced search trees, Heap and Heap sort.

**UNIT - III**

**The greedy method:** The General Method, Knapsack problem, Job Sequencing with Deadlines, Minimum- Cost Spanning Trees: Prim's Algorithm, Kruskal's Algorithm, Single Source Shortest Paths.

**Dynamic programming:** The General Method, multistage graph, Warshall's Algorithm, Floyd's Algorithm for the All-Pairs Shortest Paths Problem, single source shortest path, The Travelling Salesperson problem, optimal binary search, 0/1 knapsack.

**UNIT – IV**

**Pattern Matching:** The naïve string-matching algorithm, Brute Force String Matching, KMP algorithm.

**Np-Hard and Np-Complete Problems:** Basic concepts: non-deterministic algorithms, the classes NP - Hard and NP, NP Hard problems, clique decision problem, chromatic number decision problem, Cook's theorem.

**UNIT V**

**Backtracking:** n - Queens Problem, Subset – Sum of Subsets Problem, graph coloring.

**Branch and bound:** Assignment problem, Knapsack problem, 15 puzzle problem, travelling salesman problem

**Tries:** Standard Tries, Compressed Tries, Suffix Tries, Search Engine Indexing.

**Suggested Readings:**

1. Introduction to The Design and Analysis of Algorithms, Anany Levitin (Chapters 1-5,7,9,11), Pearson Education, Delhi, 2nd Edition, 2007, ISBN: 9780321358288.
2. Fundamentals of Computer Algorithms, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran: (Chapters 1,3-8,10-12), Universities Press, Hyderabad, 2nd Edition, 2007, ISBN: 10: 8173716129.
3. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein: PHI, London, England, 3rd Edition, 2010, ISBN: 9780262033848.
4. Introduction to the Design and Analysis of Algorithms A Strategic Approach, R.C.T. Lee, S.S. Tseng, R.C. Chang and Y.T. Tsai: cGraw-Hill Higher Education, USA, International Edition, 2005, ISBN-13: 978-0071243469.

Course Code	Course Title				Core/Elective		
U21IT501	SOFTWARE ENGINEERING				CORE		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Programming for Problem Solving	L	T	D	P			
	3	-	-	-	40	60	3
<b>Course Objectives</b> Develop ability to <ol style="list-style-type: none"> <li>1. Understand basic concepts of software development processes of defining a process framework.</li> <li>2. Impart knowledge on various phases, methodologies and practices of software development.</li> <li>3. Understand the concepts of quality standards and software change management.</li> <li>4. Understand the importance of testing in software development and study various testing strategies.</li> <li>5. Identify the risks and to study different estimation techniques.</li> </ol>							
<b>Course Outcomes</b> At the end of the course, student would be able to <ol style="list-style-type: none"> <li>1. Define different software development processes and their usability in different problem domains.</li> <li>2. Explain the process of requirements collection, analyzing, and modeling requirements for effective understanding and communication with stake holders.</li> <li>3. Design and Develop the architecture of real world problems towards developing a blue print for implementation.</li> <li>4. Understand the concepts of software equality, testing and maintenance.</li> <li>5. Discuss the concepts related to Risk management and Software project Estimation</li> </ol>							

**UNIT-I**

**Introduction to Software Engineering:** A generic view of process, Software Engineering process framework, Nature of Software, Software Myths, Process Models.

**Agile Development:** Introduction to Agility and Agile Process, Agile Process Models.

**UNIT-II**

**Understanding Requirements:** Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Building the Requirement Model, Negotiating Requirements, Validating Requirements.

**Design Concepts:** Design within the Context of Software Engineering, the Design Process, Design Concepts, Design Principles, Concept of Structural Design, DFD's, Example Data flow diagrams.

**Architectural Design:** Software Architecture, Architecture Genres, Architecture Styles, Architecture Designs.

**UNIT-III**

**Software Quality Assurance:** Background Issues, Elements of Software Quality Assurance, SQA Tasks, Goals and Metrics, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, The ISO 9000 Quality Standards, The SQA Plan.

**Risk Management:** Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management, The RMMM Plan.

**UNIT-IV**

**Testing Conventional Applications:** Software Testing Fundamentals, Internal and External Views of Testing, White-Box Testing, Basis Path Testing, Control Structure Testing, Black-Box Testing.

**Software Testing Strategies:** A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, Validation Testing, System Testing, The Art of Debugging.

## **UNIT-V**

**Software Configuration Management:** Software Configuration Management, The SCM Repository.

**Product Metrics:** A Frame work for Product Metrics, Metrics for the Requirements Model, Metrics for the Design Model, Metrics for Testing, Metrics for Maintenance.

**Estimation:** Software Project Estimation, Decomposition Techniques, Empirical Estimation Models, Specialized Estimation Techniques.

**Software Process Improvement:** The SPI Process,The CMMI.

### **Suggested Readings:**

1. Software Engineering: A Practitioners Approach, Roger S.Pressman, McGrawHill, Eighth Edition, 2019.
2. Software Engineering, Ian Somerville, Ian Somerville, Pearson Education, Eighth Edition, 2007.
3. An Integrated Approach to Software Engineering, PankajJalote,Narosa Publishing House, Third Edition 2008.

Course Code	Course Title				Core/Elective		
U21IT601	ARTIFICIAL INTELLIGENCE				CORE		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
DM, M-III	L	T	D	P			
	3	-	-	-	40	60	3
<b>Course Objectives</b> Develop ability to <ol style="list-style-type: none"> <li>1. Understand the importance of the field of AI by discussing its history and various applications.</li> <li>2. Learn about one of the basic applications of A.I, search state formulations</li> <li>3. Learn knowledge representation implementation.</li> <li>4. Learn how to reason when an agent has only uncertain information about its task.</li> <li>5. Know various supervised and unsupervised learning algorithms</li> </ol> <b>Course Outcomes</b> At the end of the course, student would be able to <ol style="list-style-type: none"> <li>1. Illustrate basic principles of AI in solutions that require problem solving, search, inference</li> <li>2. Demonstrate understanding of steps involved in building of intelligent agents, expert systems, Bayesian networks</li> <li>3. Differentiate between learning paradigms to be applied for an application</li> <li>4. Demonstrate Expert system its utilization</li> <li>5. Illustrate AI application machine learning &amp; its types.</li> </ol>							

**Unit-I:**

**Introduction to Artificial Intelligence:** Foundations of artificial intelligence (AI) ,Intelligent Systems, History of AI, Subareas of AI, Applications, Structure of Agents,Types of agents,AI problems, Agents and Environments.

**UNIT -II:**

**Basic Search Strategies:** Problem Spaces, Uninformed Search (Breadth-First, DepthFirst Search, Depth-first with Iterative Deepening), Heuristic Search (Hill Climbing, Generic Best-First, A\*), Constraint Satisfaction (Backtracking, Local Search)

**Advanced Search:** Constructing Search Trees, Stochastic Search, A\* Search Implementation, Minimax Search, Alpha-Beta Pruning.

**UNIT-III:**

**Reasoning:**Knowledge based agent, Propositional Logic, Inference, Predicate logic (first order logic),Inference,Resolution,Frames,Semantic Nets

**Uncertainty :**Basic probability, Bayes rule, Naive Bayes, Belief networks,Inference in Bayesian Network.

**UNIT-IV**

**Expert System and Applications:** Introduction, Phases in Building Expert Systems, Expert System Architecture, Applications.

**UNIT-V**

**Machine-Learning Paradigms:** Introduction, Machine Learning Systems, Supervised and Unsupervised Learning,Reinforcement learning with applications,Natural Language processing,Speech recognition.

**Suggested Readings:**

1. Artificial Intelligence – A Modern Approach, Stuart Russell and Peter Norvig, Pearson Education Press. Fourth Edition.
- 2.Artificial Intelligence, Saroj Kaushik, Cengage Learning, 2011.
- 3.Artificial Intelligence, Kevin Knight,ElaineRich,B.Nair, McGraw Hill, 3<sup>rd</sup> Edition, 2008.

Course Code	Course Title				Core/Elective		
U21IT6L1	ARTIFICIAL INTELLIGENCE LAB				CORE		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Python Programming	L	T	D	P			
	-	-	-	3	25	50	1.5
<b>Course Objectives</b> Develop ability to <ol style="list-style-type: none"> <li>1. Apply programming skills to formulate the solutions for computational problems.</li> <li>2. Implement first order predicate calculus using Prolog.</li> <li>3. Understand python library scikit-learn for building machine learning models.</li> <li>4. Select and apply relevant AI tools for the given problem.</li> <li>5. Understand the searching techniques and its implementation.</li> </ol> <b>Course Outcomes</b> At the end of the course, student would be able to <ol style="list-style-type: none"> <li>1. Develop solutions for informed and uninformed search problems in AI.</li> <li>2. Apply reasoning in first order logic using Prolog.</li> <li>3. Apply python libraries to synthesize information and develop supervised learning models.</li> <li>4. Develop a case study in multidisciplinary areas to demonstrate the use of AI.</li> <li>5. Analyse and develop real time applications by using AI tools.</li> </ol>							

### List of Experiments

1. Python program to implement List operations.
2. Implementation of Nested List,Length ,Concatenation, Membership, Iteration, Indexing and Slicing functions.
3. Implementation of List methods : Add, Append, Extend & Delete.
4. Program to eliminate punctuations in a given string.
5. Sorting a sentence in an alphabetical order.
6. Implementation of Breadth First Search Traversal Technique.
7. Depth First Search Traversal Technique Implementation.
8. Implementation of Water Jug Problem using Heuristic Method.
9. Implementation of Travelling Salesman Problem using Python.
10. Writing simple facts for the statements and querying using Prolog.
11. Demonstration of Family tree using Prolog using conditional statements.
12. CASE STUDY
  - a. Simple Chatroom using Python
  - b.Hospital Management System Project in Python

### Suggested Readings:

1. Artificial Intelligence – A Modern Approach, Stuart Russell and Peter Norvig, Pearson Education Press, Third edition.
2. Artificial Intelligence, Saroj Kaushik, Cengage Learning, 2011.

Course Code	Course Title				Core/Elective		
U21IT5L1	SOFTWARE ENGINEERING LAB				CORE		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Programming for Problem Solving	L	T	D	P			
	-	-	-	3	25	50	1.5
<p><b>Course Objectives</b> Develop ability to</p> <ol style="list-style-type: none"> <li>1. Have an experience on identifying the problem statement and eliciting the requirements of software.</li> <li>2. Design a high level model of system by requirements provided by customer.</li> <li>3. Study and use different Design Phase CASE tools.</li> <li>4. Identify risks and change management of software product.</li> <li>5. Apply different testing strategies to test the software.</li> </ol> <p><b>Course Outcomes</b> At the end of the course, student would be able to</p> <ol style="list-style-type: none"> <li>1. Translate end-user requirements into system and software requirements</li> <li>2. Generate a high-level design of the system from the software requirements</li> <li>3. Identify the risks associated with the software developed.</li> <li>4. Design the test case to test the software developed.</li> <li>5. Experience of testing problems and will be able to develop a simple testing report.</li> </ol>							

### List of Experiments

Do the following 7 exercises for any two projects given in the list of sample projects or any other projects using Agile Model.

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

### Suggested Readings:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, Eighth Edition, Mc Graw Hill International Edition.
2. Software Engineering- Ian Sommerville, Eighth Edition, Pearson Education.
3. The Unified Modeling Language User Guide, Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

Course Code	Course Title				Core/Elective		
U21IT5P1	INTERNSHIP				CORE		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Programming Languages	L	T	D	P			
	-	-	-	2	50	-	1
<p><b>Course Objectives</b> Develop ability to</p> <ol style="list-style-type: none"> <li>1. Produce an accurate record of work performed during the Internship/Co-op</li> <li>2. Apply engineering knowledge to a problem in industry</li> <li>3. Produce a technical report</li> <li>4. Discuss work in a team environment, if relevant to the project</li> <li>5. Conduct herself/himself responsibly, safely, and ethically in a professional environment</li> </ol> <p><b>Course Outcomes</b> At the end of the course, student would be able to</p> <ol style="list-style-type: none"> <li>1. Design and develop a small and simple product in hardware or software.</li> <li>2. Complete the task or realize a pre specified target, with a specified scope.</li> <li>3. Learn to find alternate viable solutions for a given problem and evaluate these alternatives with reference to pre specified criteria.</li> <li>4. Gain knowledge of working practives withing industrial / R&amp;D environments.</li> <li>5. Implement the selected solution and document the same.</li> </ol>							

### Guidelines:

Internship is introduced as part of the curriculum for encouraging students to work on problems of interest to industries. A batch of three students will be attached to a person from the Government or Private Organisations /Computer Industry/Software Companies/R&D Organization for a period of 4 weeks. This will be during the summer vacation following the completion of the IV Semester. One faculty coordinator will also be attached to the group of 3 students to monitor the progress and to interact with the industry coordinate (person from industry). The course schedule will depend on the specific internship/training experience. The typical time per topic will vary depending on the internship

1. Overview of company/project
2. Safety training
3. Discussions with project teams
4. Background research, review of documents, white papers, and scientific papers
5. Planning, designing, and reviewing the planned work
6. Executing the plans
7. Documenting progress, experiments, and other technical documentation
8. Further team discussions to discuss results
9. Final report writing and presentation

After the completion of the internship, each student will be required to:

1. Submit a brief technical report on the internship undergone.
2. Present the work through a seminar talk (to be organized by the Department)

Internship will be evaluated for 50 marks as CIE based on Punctuality in Attendance (10), Maintenance of Diary on daily work done (20), seminar presentation / viva voce followed by a report submission (20) to a committee consisting of Head of the Department, Senior Faculty and Faculty Advisor / Supervisor for evaluation.

**Note:** Students have to undergo summer internship of 4 weeks at the end of semester IV and credits will be awarded after evaluation in V semester.

Course Code	Course Title				Core/Elective		
U21MA5L1	APTITUDE AND REASONING				BSC		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	-	-	-	2	25	50	1
<p><b>Course Objectives</b> Develop ability to</p> <ol style="list-style-type: none"> <li>1. Acquire the concepts of mathematical aptitude and reasoning.</li> <li>2. Develop the innovative and creative thinking through basic mathematical concepts.</li> <li>3. Improve analytical and problems solving skills.</li> <li>4. Enhance logical thinking and mathematical ability.</li> <li>5. Empower with the basic tools of mathematical aptitude.</li> </ol> <p><b>Course Outcomes</b> At the end of the course, student would be able to</p> <ol style="list-style-type: none"> <li>1. Build proficiency in quantitative reasoning.</li> <li>2. Improve critical thinking skills.</li> <li>3. Enhance analytical skills.</li> <li>4. Demonstrate quantitative aptitude concepts.</li> <li>5. Adapt principles of quantitative aptitude to achieve qualitative results.</li> </ol>							

**List of Experiments:**

- 1) **Basic concepts:** combined mean, average principles, wrong values taken, number added or deleted, average speed.
  - 2) **Percentages** -Basic Concepts, conversions, finding percentages from given numbers, quantity increases or decreases by given percentage, population increase by given percentage, comparisons, consumption when a commodity price increase or decrease and applications.
  - 3) **Data Interpretation** - Introduction to Data Interpretation, quantitative and qualitative data, Tabular Data, Line Graphs, Bar Chart, Pie Charts, X-Y Charts.
  - 4) Number Series, Letter Series, Series completion and correction, Coding and Decoding. Word analogy Applied analogy, Classifications, verbal classification.
  - 5) **Reasoning Logical Diagrams** - Simple diagrammatic relationship, Multi diagrammatic relationship, Venndiagrams, Analytical reasoning.
- Reasoning Ability** - Blood Relations, Seating arrangements, Directions, Decision making.
- Number Systems:** Basic Concepts.
- 6) **Number Systems:** Natural numbers, whole numbers, integers, fractions, Rational Numbers, Irrational Numbers, Real Numbers, Divisibility Rules, Logic Equations, Remainder theorem, Unit digit calculation. Progressions & Inequalities: Basic Concepts, Types: arithmetic, geometric, harmonic progression and applications.
  - 7) **Profit and Loss:** Basic Concepts, discounts, marked price and list price, dishonest shopkeeper with manipulated weights, successive discounts etc.  
Interest (Simple and Compound): Basic Concepts, Yearly, Half-yearly, and quarterly calculations, multiples, differences between simple and compound interest.
  - 8) **Ratio and Proportion:** Basic Concepts of ratio and proportion, continued or equal proportions, mean proportions, invest proportion, alternative proportion, division proportion, compound proportion, duplication of ratio, finding values, coins and currencies, etc.
  - 9) **Speed, Time and Distance:** Basic Concepts, Single train problems, two train problems: some point same side, some point opposite sides, relative speed, different points meeting at common points, different points same side (different timings vs. same timings), ratios, number of stoppages, average speed, etc.
  - 10) **Time and Work:** Basic Concepts, comparative work, mixed work, alternative work, middle leave and middle join, ratio efficiency.

11) **Permutations and combinations:** Basic Concepts, differences between permutations and combinations, alternative arrangement, fixed positions, items drawing from a single group, items drawing from a multiple group, total ways of arrangement with repetitions and without repetitions, handshakes or line joining between two points or number of matches, sides and diagonals, etc.

12) **Clocks and Calendars:** Basic Concepts, Angle between minute hand and hour hand, reflex angle, hours hand angle, time gap between minute hand and hour hand, relative time: coincide, opposite sides and right angle, mirror images, faulty clock (slow/fast), miscellaneous, calendar.

13) **Geometry and Mensuration:** Basic concepts, types of angles.

Plane figures: rectangles, squares, triangles, quadrilateral, areas, perimeters, etc.

Solid figures: cubes, cuboids, cylinders-area (total surface area and lateral surface area), volumes, perimeters.

**Others:** Parallelogram, Rhombus, Trapezium, Circle, Sector, Segment, Cone, Sphere, Hemisphere, etc.

**Suggested Readings:**

1. Aptitude and reasoning skills lab Manual, LIET, HYD.

Course Code	Course Title				Core/Elective		
U21IT602	COMPUTER NETWORKS				CORE		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Operating Systems	L	T	D	P			
	3	-	-	-	40	60	3
<b>Course Objectives:</b> Develop ability to <ol style="list-style-type: none"> <li>1. Become familiar with layered communication architectures (OSI and TCP/IP).</li> <li>2. Familiarize with flow control and error control mechanisms in data link layer</li> <li>3. Study the design issues in network layer and various routing algorithms</li> <li>4. Learn sockets programming and how to implement client/server programs.</li> <li>5. Understand the concepts of reliable data transfer and how TCP implements these concepts</li> </ol>							
<b>Course Outcomes:</b> At the end of the course, student would be able to <ol style="list-style-type: none"> <li>1. Explain the function of each layer of OSI and trace the flow of information from one Node to another node in the network</li> <li>2. Familiarise with the Transmission Media, Flow Control and Error Detection and Correction</li> <li>3. Understand the principles of IP addressing and internet routing</li> <li>4. Describe the working of various networked applications such as DNS, mail, file transfer and www</li> <li>5. Implement client-server socket-based networked applications.</li> </ol>							

### Unit-I

**Introduction:** Network Uses, Topologies, Transmission Modes, Types of computer networks, Examples of Networks, Network Technology-from local to global, Reference Models- OSI, TCP/IP.

**The Physical Layer:** Transmission Media, Guided transmission media, Twisted Pair, Coaxial Cable, Fiber Optics, Wireless transmission- Electromagnetic Spectrum, Frequency Hopping Spread Spectrum, Direct Sequence Spread Spectrum, Ultra Wide Band Connection, Cellular Networks-Common Concepts, Technology, Analog Voice, Digital Voice, GSM, Digital Voice and Data, Packet Switching and 5G technology.

### Unit-II

**The Data Link Layer:** Design Issues, Error Detection and Correction-Error Detecting Codes, Error Correcting Codes, Elementary Data Link Protocols-Initial Simplifying Assumptions, Basic Transmission and receipts, Simplex Link Layer Protocols.

**The Medium Access Sublayer:** Channel allocations problem- Static Channel Allocation, Assumptions for Dynamic Channel Allocation, multiple access protocols- CSMA, Collision Free Protocols, Wireless LAN Protocols, Ethernet, Wireless LAN.

### Unit-III

**The Network Layer:** Network layer design issues, Routing Algorithms- Distance Vector Routing, Link State Routing, Hierarchical Routing, Broadcast Routing, Multicast Routing.

**Internetworking:** Concatenated virtual circuits, Connectionless internet working, Tunneling, The network layer in the internet-IP protocol, IP addresses, OSPF, BGP, (IPv4 and IPv6), Internet Control Protocols- ARP, RARP, ICMP, DHCP.

### UNIT-IV

**Network Programming:** Socket Interface- Sockets, Socket Address, Elementary Sockets, Advanced Sockets.

**The Transport Layer:** Transport service, elements of transport protocol- Addressing, Connection Establishment, Connection Release, Multiplexing, UDP- Introduction, RPC, TCP- Introduction, TCP Service Model, TCP Protocol, Segment Header, Connection Establishment and Connection release.

### **UNIT-V**

**The Application Layer:** Domain Name System, Electronic Mail-Architecture and Services, World Wide Web-architectural overview, dynamic web document, http, HTTPS, Web Privacy, Content Delivery-Content and Internet Traffic, Server Farms and Web Proxies, Peer –to- Peer Networks, Evolution of the Internet.

#### **Suggested Readings:**

- 1 Computer Networks, Andrew S.Tanenbaurn, Pearson Education, Sixth Edition, 2021.
- 2.Unix Network Programming, W. Richard stevens, Prentice Hall/PearsonEducation, 2009.
- 3.Computer Networking: A Top-Down Approach Featuring the Internet, JamesF.Kurose, KeithW,Ross, PearsonEducation, Third Edition, 2005.
- 4.Data Communications and Networking with TCP/IP Protocol Suite, Behrouz A. Forouzan, McGraw Hill LLC, Sixth Edition, 2021.

Course Code	Course Title				Core/Elective		
U21CD601	MACHINE LEARNING				CORE		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Python Programming	L	T	D	P			
	3	-	-	-	40	60	3
<b>Course Objectives</b> Develop ability to <ol style="list-style-type: none"> <li>1. Understand the basic concepts of machine learning and range of problems that can be handled by machine learning</li> <li>2. Analyze the concepts learning and decision tree induction</li> <li>3. Understand the concepts of unsupervised learning and evolutionary learning</li> <li>4. Understand the concepts of ensemble learning, dimensionality reduction and clustering</li> <li>5. Gain experience of doing independent study and research</li> </ol> <b>Course Outcomes</b> At the end of the course, student would be able to <ol style="list-style-type: none"> <li>1. Design and implement machine learning solutions imply variant techniques,</li> <li>2. Calculate Statistical measurements of the given data.</li> <li>3. Analyze and identify the best algorithm matches for a given dataset.</li> <li>4. Evaluate and interpret the results of the machine learning algorithms</li> <li>5. Design and implement various machine learning algorithms in a range of real-world applications</li> </ol>							

**Unit - I**

**Introduction:** Learning, Types of Machine Learning.

**Concept learning:** Introduction, Version Spaces and the Candidate Elimination Algorithm.

**Some Basic Statistics:** Averages, Variance and Covariance, The Gaussian, The Bias-Variance, Bayes theorem. Bayes Optimal Classifier, Naïve Bayes Classifier.

**Unit - II**

**Learning with Trees:** Constructing Decision Trees, CART, Classification Example

**Linear Discriminants:** The Perceptron, Linear Separability, Linear Regression

**Multilayer Perceptron (MLP):** Going Forwards, Backwards, MLP in practices, Deriving back.

**Unit - III**

**Clustering:** Introduction, Similarity and Distance Measures, Outliers, Hierarchical Methods, Partitional Algorithms, Clustering Large Databases, Clustering with Categorical Attributes, Comparison

**Graphical Models:** Bayesian networks, Approximate Inference, Making Bayesian Networks, HiddenMarkov Models, The Forward Algorithm.

**Unit - IV**

**Evolutionary Learning:** Genetic Algorithms, Genetic Operators, Genetic Programming.

**Ensemble learning:** Boosting, Bagging.

**Dimensionality Reduction:** Linear Discriminant Analysis, Principal Component Analysis

**Unit – V**

**Reinforcement Learning:** Introduction, the learning task, Q-learning, non-deterministic, rewards and actions, temporal difference learning, generalizing from examples, relationship to dynamic programming.

**Suggested Readings:**

1. Machine Learning - An Algorithmic Perspective, Stephen Marsland, CRC Press, 2009
2. Machine Learning, Tom M. Mitchell, Mc Graw Hill, 1997
3. Data Mining, Margaret H Dunham, Pearson Edition., 2003.
4. Data Mining for Business Intelligence, GalitShmueli, Nitin R Patel, Peter C Bruce, Wiley India Edition, 2007
5. Pattern Recognition, RajjanShinghal, Oxford University Press, 2006.

Course Code	Course Title				Core/Elective		
U21IT6L2	COMPUTER NETWORKS LAB				CORE		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Operating Systems	L	T	D	P			
	-	-	-	3	25	50	1.5
<p><b>Course Objectives</b> Develop ability to</p> <ol style="list-style-type: none"> <li>1. Understand the use of client/server architecture in application development.</li> <li>2. Understand and use elementary socket system calls in TCP &amp; UDP based sockets.</li> <li>3. Implement and use advanced socket system calls and TCP and UDP based sockets.</li> <li>4. Work with the use of concurrent server services.</li> <li>5. Simulate chat applications using TCP &amp; UDP.</li> </ol> <p><b>Course Outcomes</b> At the end of the course, student would be able to</p> <ol style="list-style-type: none"> <li>1. Understand the usage of basic commands ipconig, ifconfig, netstat, ping, arp, telnet, ftp, finger, trace route, who is of LINUX platform.</li> <li>2. Develop and Implement Client-Server Socket based programs using TCP, and UDP sockets.</li> <li>3. Make a client server communication through TCP and UDP protocols.</li> <li>4. Expose on advanced socket programming in LINUX environment.</li> <li>5. Understand transport layer protocols, connection oriented &amp; connectionless models.</li> </ol>							

### List of Experiments

1. Familiarization of Network Environment, Understanding and using network utilities: ipconig, ifconfig, netstat, ping,arp,telnet, ftp,finger, traceroute,whois.
2. Implementation of connection oriented and connection less client and server for well-known services i.e., standard ports.
3. Implementation of concurrent server service using connection-oriented socket system calls(Service: Daytime)
4. Implementation of concurrent server using connection less socket system calls.(Service: Echo server)
5. Implementation of Client Communication with TCP chat server
6. Implementation of Client Communication with UDP chat server
7. Program to demonstrate the use of advanced socket system calls: readv(), writev(), getsockname(), setsockname(), getpeername().

### Suggested Readings:

1. Unix Network Programming, W.RichardStevens,Prentice Hall, Pearson Education,2009.
2. Handson Networking with Internet Technologies, Douglas E.Comer,Pearson Education.

Course Code	Course Title				Core/Elective		
U21CD6L2	<b>MACHINE LEARNING LAB</b>				<b>CORE</b>		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
	-	-	-	3	25	50	1.5
<p><b>Course Objectives</b> Develop ability to</p> <ol style="list-style-type: none"> <li>1. Make use of Data sets in implementing the machine learning algorithms</li> <li>2. Implement the machine learning concepts and algorithms in any suitable language of choice.</li> </ol> <p><b>Course Outcomes</b> At the end of the course, student would be able to</p> <ol style="list-style-type: none"> <li>1. Understand the implementation procedures for the machine learning algorithms.</li> <li>2. Design Java/Python programs for various Learning algorithms.</li> <li>3. Apply appropriate data sets to the Machine Learning algorithms.</li> <li>4. Identify and apply Machine Learning algorithms to solve real world problems.</li> </ol>							

### List of Experiments

1. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
2. For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.
3. Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
4. Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.
5. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
6. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision, and recall for your data set.
7. Write a program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set. You can use Java/Python ML library classes/API.
8. Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.
9. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.
10. Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

Course Code	Course Title				Core/Elective		
U21IT5L2	SCRIPTING LANGUAGES LAB				CORE		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Java	L	T	D	P			
	-	-	-	2	50	-	1
<p><b>Course Objectives</b> Develop ability to</p> <ol style="list-style-type: none"> <li>1. Develop an ability to design and implement static and dynamic website</li> <li>2. Create conforming web pages</li> <li>3. Use JavaScript for dynamic effects</li> <li>4. Understand, analyze and create XML documents and XML Schema</li> <li>5. Use appropriate client-side or Server-side applications.</li> </ol> <p><b>Course Outcomes</b> At the end of the course, student would be able to</p> <ol style="list-style-type: none"> <li>1. Understand, analyze and apply the role of languages like HTML, CSS, XML, JavaScript, PHP, SERVLETS, JSP and protocols in the workings of the web and web applications</li> <li>2. Design and implement dynamic websites with good aesthetic sense of designing</li> <li>3. Create web pages using HTML and Cascading Styles sheets</li> <li>4. Analyze a web page and identify its elements and attributes.</li> <li>5. Develop JSP applications implementing Session management and Data base Connectivity.</li> </ol>							

### List of Experiments

- 1) Design the following static web pages required for an online book store web site.
  - a) HOME PAGE: The static home page must contain three frames.
  - b) LOGIN PAGE
  - c) CATALOGUE PAGE: The catalogue page should contain the details of all the books available in the web site in a table.
  - d) REGISTRATION PAGE.
- 2) Write JavaScript to validate the following fields of the Registration page.
  1. First Name (Name should contains alphabets and the length should not be less than 6 characters).
  2. Password (Password should not be less than 6 characters length).
  3. E-mail id (should not contain any invalid and must follow the standard pattern name@domain.com)
  4. Mobile Number (Phone number should contain 10 digits only).
  5. Last Name and Address (should not be Empty).
- 3) Develop and demonstrate the usage of inline, internal and external style sheet using CSS
- 4) Develop and demonstrate JavaScript with POP-UP boxes and functions for the following problems:
  - a) Input: Click on Display Date button using onclick( ) function Output: Display date in the textbox
  - b) Input: A number n obtained using prompt Output: Factorial of n number using alert
  - c) Input: A number n obtained using prompt Output: A multiplication table of numbers from 1 to 10 of n using alert
  - d) Input: A number n obtained using prompt and add another number using confirm Output: Sum of the entire n numbers using alert.
- 5) Write an HTML page that contains a selection box with a list of 5 countries. When the user selects a country, its capital should be printed next in the list. Add CSS to customize the properties of the font of the capital (color, bold and font size)
- 6) Write an HTML page including any required JavaScript that takes a number from text field in the range of 0 to 999 and shows it in words. It should not accept four and above digits, alphabets and special characters.

- 7) Develop and demonstrate PHP Script for the following problems:
- Write a PHP Script to find out the Sum of the Individual Digits.
  - Write a PHP Script to check whether the given number is Palindrome or not.
- 8) Create an XML document that contains 10 users information. Write a Java Program, which takes User Id as input and returns the user details by taking the user information from XML document using DOM parser or SAX parser.
- 9) Implement the following web applications using (a) PHP (b) JSP
- A web application that takes a name as input and on submit it shows a hello page where name is taken from the request. It shows the start time at the right top corner of the page and provides a logout button. On clicking this button, it should show a logout page with Thank You message with the duration of usage (hint: Use session to store name and time).
  - Write a PHP Program to display current Date, Time and Day.
- 10) Implement the following web applications using (a) PHP (b) JSP
- A web application that takes name and age from an HTML page. If the age is less than 18, it should send a page with "Hello , you are not authorized to visit the site" message, where should be replaced with the entered name. Otherwise it should send "Welcome to this site" message.
  - A web application that lists all cookies stored in the browser on clicking "List Cookies" button. Add cookies if necessary.
- 11) Implement the web applications with Database using (a) PHP, JSP
- 12) Write a program to design a simple calculator using (a) JavaScript (b) PHP (c) JSP.

**Suggested Readings:**

- The Complete Reference, StevenHolzner, PHP,TataMcGraw-Hill,1<sup>st</sup> Edition,2007
- Web Technologies, OxfordUniversityPress, UttamKRoy, 1<sup>st</sup> Edition,2010.
- Java Script: The Definitive Guide, David Flanagan O'Reilly, 6<sup>th</sup> Edition

Course Code	Course Title				Core/Elective		
U21IT6P1	MINI PROJECT				CORE		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Programming Languages	L	T	D	P			
	-	-	-	6	50	50	3
<p><b>Course Objectives</b> Develop ability to</p> <ol style="list-style-type: none"> <li>1. Enhance practical and professional skills.</li> <li>2. Familiarize tools and techniques of systematic literature survey and documentation</li> <li>3. Expose the students to industry practices and team work.</li> <li>4. Encourage students to work with innovative and entrepreneurial ideas.</li> <li>5. Make students evaluate different solutions based on economic and technical feasibility</li> </ol> <p><b>Course Outcomes</b> At the end of the course, student would be able to</p> <ol style="list-style-type: none"> <li>1. Formulate a specific problem and give solution</li> <li>2. Develop model/models either theoretical/practical/numerical form</li> <li>3. Solve, interpret/correlate the results and discussions</li> <li>4. Conclude the results obtained</li> <li>5. Write the documentation in standard format</li> </ol>							

### Guidelines:

1. As part of the curriculum in the VI- semester of the programme each student shall do a mini project, generally comprising about three to four weeks of prior reading, twelve weeks of active research, and finally a presentation of their work for assessment.
2. Four students will be allotted to one faculty supervisor for mentoring.
3. Mini projects should present students with an accessible challenge on which to demonstrate competence in research techniques, plus the opportunity to contribute something more original.
4. Mini projects shall have inter-disciplinary/ industry relevance.
5. The students can select a mathematical modelling based/Experimental investigations or Numerical modelling
6. All the investigations should be clearly stated and documented with the reasons/explanations.
7. The mini-project shall contain a clear statement of the research objectives, background of work, literature review, techniques used, prospective deliverables, and detailed discussion on results, conclusions and reference.
8. The CIE shall include reviews and the preparation of report consisting of a detailed problem statement and a literature review.
9. The preliminary results (if available) of the problem may also be discussed in the report.
10. The work has to be presented in front of the PRC committee which consists of one Supervisor and a minimum of two faculty members from the respective Department of the Institute.

Course Code	Course Title				Core/Elective		
U21IT604	<b>PRINCIPLES OF PROGRAMMING LANGUAGES</b>				<b>Professional Elective-I</b>		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Basics of Computer	L	T	D	P			
	3	-	-	-	40	60	3
<b>Course Objectives</b> Develop ability to <ol style="list-style-type: none"> <li>1. Learn importance of programming languages paradigm.</li> <li>2. Understand concepts of high-level language design and implementation.</li> <li>3. Work with programming syntax and semantics, data types, expressions and statements.</li> <li>4. Understand subprograms and programming blocks.</li> <li>5. Differentiate abstract data types, functional and logic programming languages.</li> </ol> <b>Course Outcomes</b> At the end of the course, student would be able to <ol style="list-style-type: none"> <li>1. Acquire the skills for expressing syntax and semantics in formal notation.</li> <li>2. Identify and apply a suitable programming paradigm for a given computing application</li> <li>3. Understand a program using functions and work with abstract data types.</li> <li>4. Work with the concurrency management systems.</li> <li>5. Work with the functional programming languages and scripting languages.</li> </ol>							

**UNIT – I**

**Preliminary Concepts:** Reasons for Studying Concepts of Programming Languages, Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Trade-Offs, Implementation Methods, Programming Environments Syntax and Semantics: General Problem of Describing Syntax and Semantics, Formal Methods of Describing Syntax, Attribute Grammars, Describing the Meanings of Programs.

**UNIT – II**

**Names, Bindings, and Scopes:** Introduction, Names, Variables, Concept of Binding, Scope, Scope and Lifetime, Referencing Environments, Named Constants Data Types: Introduction, Primitive Data Types, Character String Types, User Defined Ordinal Types, Array, Associative Arrays, Record, Union, Tuple Types, List Types, Pointer and Reference Types, Type Checking, Strong Typing, Type Equivalence Expressions and Statements: Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short Circuit Evaluation, Assignment Statements, Mixed-Mode Assignment Control Structures – Introduction, Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands.

**UNIT – III**

**Subprograms and Blocks:** Fundamentals of Sub-Programs, Design Issues for Subprograms, Local Referencing Environments, Parameter Passing Methods, Parameters that Are Subprograms, Calling Subprograms Indirectly, Overloaded Subprograms, Generic Subprograms, Design Issues for Functions, User Defined Overloaded Operators, Closures, Co-routines Implementing Subprograms: General Semantics of Calls and Returns, Implementing Simple Subprograms, Implementing Subprograms with Stack-Dynamic Local Variables, Nested Subprograms, Blocks, Implementing Dynamic Scoping Abstract Data Types: The Concept of Abstraction, Introductions to Data Abstraction, Design Issues, Language Examples, Parameterized ADT, Encapsulation Constructs, Naming Encapsulations.

#### **UNIT – IV**

**Concurrency:** Introduction, Introduction to Subprogram Level Concurrency, Semaphores, Monitors, Message Passing, Java Threads, Concurrency in Function Languages, Statement Level Concurrency. Exception Handling and Event Handling: Introduction, Exception Handling in Ada, C++, Java, Introduction to Event Handling, Event Handling with Java and C#.

#### **UNIT – V**

**Functional Programming Languages:** Introduction, Mathematical Functions, Fundamentals of Functional Programming Language, LISP, Support for Functional Programming in Primarily Imperative Languages, Comparison of Functional and Imperative Languages Logic Programming Language: Introduction, an Overview of Logic Programming, Basic Elements of Prolog, Applications of Logic Programming. Scripting Language: Pragmatics, Key Concepts, Case Study: Python – Values and Types, Variables, Storage and Control, Bindings and Scope, Procedural Abstraction, Data Abstraction, Separate Compilation, Module Library.

#### **Suggested Readings:**

1. Programming Languages, A.B. Tucker, R. E. Noonan, TMH. 2nd Edition.
2. Programming Languages, K. C. Loudon, Thomson, 2nd Edition, 2003

Course Code	Course Title				Core/Elective		
U21IT605	DATA MINING				Professional Elective-I		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
DBMS	L	T	D	P			
	3	-	-	-	40	60	3
<b>Course Objectives</b> Develop ability to <ol style="list-style-type: none"> <li>1. Understand data classification, data preprocessing and data mining applications.</li> <li>2. Work with the patterns, associations and correlations can be obtained on data.</li> <li>3. Describe various classification techniques.</li> <li>4. Understand the classification and clustering techniques can be implemented &amp; perform its evaluation.</li> <li>5. Learn how complex data mining can be performed.</li> </ol> <b>Course Outcomes</b> At the end of the course, student would be able to <ol style="list-style-type: none"> <li>1. Classify types of data, perform preprocessing of data and appreciate applications of data mining.</li> <li>2. Analyze data for mining frequent patterns, Associations and Correlations.</li> <li>3. Perform the classification by using decision tree induction, Bayes classification methods etc. and evaluate the classifier.</li> <li>4. Select and perform clustering, outlier analysis detection methods.</li> <li>5. Perform Text mining, Spatial Mining, Web mining and Multimedia mining.</li> </ol>							

**UNIT-I**

**Introduction:** Fundamentals of Data Mining, Kinds of Patterns can be mined, Technologies used, Applications and issues in Data Mining. Types of Data: Attribute types, Basic Statistical Descriptions of Data, Measuring data similarity and Dissimilarity. Data Pre-Processing: Need of Pre-processing, Data Cleaning, Data Integration, Data Reduction, Data Transformation.

**UNIT-II**

**Mining Frequent Patterns, Associations and Correlations:** Market Basket Analysis, Association rule mining, frequent item set mining methods, mining various kinds of association rule, Constraint based frequent pattern mining.

**UNIT-III**

**Classification:** General approach to classification, Classification by Decision tree induction, Classification by back Propagation, Lazy learners, other classification methods, Prediction, Evaluating the accuracy of classifier, increasing the accuracy of classifier.

**UNIT-IV**

**Cluster Analysis:** Basic Clustering methods, Partitioning methods, Density-based methods, Grid-based methods, and Evaluation of clustering, Outlier Analysis and detection methods.

**UNIT-V**

**Mining Complex Data, Applications and Trends:** Mining complex data: Spatial mining, Text Mining, Multimedia Mining, Web Mining, Data Mining Applications and Data Mining Trends.

**Suggested Readings:**

1. Data Mining: Concepts and Techniques, HanJ & Kamber M, Harcourt India, Elsevier India, Second Edition.
2. Introduction to Data Mining, Pang-Ning Tan. Michael Steinback, Vipin Kumar, Pearson Education, 2008.
3. Data Mining: Introductory and Advanced Topics, Margaret H Dunham, S. Sridhar, Pearson Education, 2008.
4. Data Warehousing: Architecture and Implementation, Humphires, Hawkins, Dy. Pearson Education, 2009.
5. Data Warehousing in the Real World, Anahory, Murray, Pearson Education, 2008.

Course Code	Course Title				Core/Elective		
U21IT606	ADVANCED OPERATING SYSTEMS				Professional Elective-I		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Operating Systems	L	T	D	P			
	3	-	-	-	40	60	3
<b>Course Objectives</b> Develop ability to <ol style="list-style-type: none"> <li>1. Get a comprehensive knowledge of the architecture of distributed systems and Real-Time operating system.</li> <li>2. Understand the deadlock and shared memory issues and their solutions in distributed environments.</li> <li>3. Get a knowledge on Multiprocessor Operating System and Database Operating System.</li> <li>4. Give an understanding of practical engineering issues in real-time and concurrent systems and suggest appropriate implementations techniques.</li> <li>5. Know the security issues and protection mechanisms for distributed environments.</li> </ol> <b>Course Outcomes</b> At the end of the course, student would be able to <ol style="list-style-type: none"> <li>1. Analyze a variety of real-time scheduling techniques.</li> <li>2. Understand the main concepts of resource management techniques for distributed systems and the Kernel Based Approach.</li> <li>3. Understand the features of database operating system.</li> <li>4. Analyze the characteristics of Multiprocessor Operating system.</li> <li>5. Compare and evaluate different programming model for concurrent system, their implementation and their impact on operating system.</li> </ol>							

**UNIT-1**

**Process Synchronization:** Overview, Functions of an Operating System, Design Approaches, Why Advance Operating Systems, Types of Advanced Operating System.

**Synchronization Mechanism:** Introduction, Concept of a Process, concurrent Process, The Critical-Section Problem, other synchronization problem, language mechanisms for synchronization.

**Process Deadlocks:** Introduction, Preliminaries, Models of deadlock, Models of resources, A Graph theoretical model of a system State, System with single unit request, systems with only consumable request, Systems with only reusable resources.

**UNIT-2**

**Distributed Operating System:** introduction, System Architectures types, Distributed Operating System, Issues in Distributed operating system, Communication networks, communication Primitives.

**Theoretical Foundations:** Introduction, Inherent Limitations of a Distributed Systems, Lamport's logical Clock, Vector clocks, Casual Ordering Messages, Global State,

**Distributed Mutual Exclusion:** Introduction, The classification of Mutual Exclusion Algorithm, Preliminaries, A solution to distributed Mutual Exclusion, non-Token Based Algorithms, Lamport's algorithm, the Ricart-Agrawala Algorithm, Maekawa's Algorithm, Token based algorithm, Suzuki-Kasami's broadcast algorithm, Singhal's heuristics Algorithm, Raymond's tree based Algorithm.

**UNIT-3**

**Multiprocessor Operating System:** Introduction, Structures of Multiprocessor operating system, Operating System Design issues, Threads, Process Synchronization, Processor Scheduling, Memory Management- The Mach-operating system, Reliability/Fault Tolerance: The Sequoia System.

**Multiprocessor System Architectures:** Introduction, Basic multiprocessor system architecture, interconnection networks for multiprocessor system, Caching, Hyper cube architecture.

**UNIT-4**

**Database Operating System:** Introduction, Concurrency Control, Distributed Database System, Concurrency Control Algorithm-Lock Based Algorithm, Timestamp Based Algorithm, optimistic Algorithm, Data Replications

**Real Time operating System:** Overview, System Characteristics, Features of Real time Kernel, Implementing Real time Operating System, Real Time CPU-Scheduling,

#### **UNIT-5**

**Protection and Security:** Resource security and Protection: Access and Flow Control: Introduction, The Access Matrix Model, Implementations of Access matrix, Safety in Access Matrix Model, Advance Models of Protection, Case Study: The Hydra Kernel.

**Data Security Cryptography:** Introduction, A model of cryptography, Conventional Cryptography, Modern Cryptography, Private and public Key Cryptography, Multiple Encryption, Case Study-The Kerberos System.

#### **Suggested Readings:**

1. Advance concepts in Operating Systems by Mukesh Singhal, Niranjan G. Shivratri, Mc Graw Hill Education, 2001.
2. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System principles, Seventh Edition, John wiley&sons publication,2012

Course Code	Course Title					Core/Elective	
U21IT607	<b>SOFTWARE TESTING METHODOLOGIES</b>					<b>Professional Elective-I</b>	
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
Software Engineering	L	T	D	P			
	3	-	-	-	40	60	3
<b>Course Objectives</b> Develop ability to <ol style="list-style-type: none"> <li>1. Understand and learn the basic concepts of Testing.</li> <li>2. Follow the methodology of White Box Testing.</li> <li>3. Obtain knowledge of Integration and System Testing</li> <li>4. Understand the concepts of Object-Oriented Testing.</li> <li>5. Gain the knowledge of complex problems</li> </ol> <b>Course Outcomes</b> At the end of the course, student would be able to <ol style="list-style-type: none"> <li>1. Gain the basic knowledge of Testing.</li> <li>2. Acquire the knowledge of White Box Testing methods</li> <li>3. Test an application using Functional Testing.</li> <li>4. Use Object Oriented Testing and Millennium Testing methods</li> <li>5. Solve software complexities with best practices.</li> </ol>							

**UNIT - I**

**A perspective on testing:** Basic definitions, Test cases, Insights from a Venn diagram, Identifying test cases, Error and fault taxonomies, Verification, Validation and Testing, Levels of testing. Examples: Generalized pseudo code, The triangle problem, The NextDate function, The commission problem, The SATM (Simple Automatic Teller Machine) problem, The currency converter, Saturn windshield wiper

**UNIT – II**

**Life Cycle – based testing:** Water fall model– V–model– Spiral model– Agile model, Life cycle of testing, Static Testing dynamic testing, Code coverage testing, Code complexity testing, White box testing Block box testing, Regression testing, Integration Testing ,System and Performance Testing

**UNIT - III**

**Path testing:** Paths, Path products and Regular expressions, Path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection, Boundary Value Analysis, Data Flow Testing, Slice based testing.

**UNIT - IV**

**Test planning:** Introduction, cost–benefit analysis of testing, monitoring and control, test reporting, test control Specialized testing, Object Oriented Testing.

**UNIT – V**

**Testing tools:** Automated tools for testing, Tool Selection & Implementation, Challenges in test automation, GUI Testing, Software Technical Reviews, Software Testing Excellence- Best Practices.

**Suggested Readings:**

1. Software Testing: A Craftsman’s Approach, Paul C. Jorgensen, CRC Press, 4th Edition, 2007
2. Software Testing Techniques, Boris Biezer, Dreamtech Press, 2nd Edition, 2003
3. Software Testing Tools, Dr. K.V. K. K. Prasad, Dreamtech.
4. Foundations of Software Testing, Aditya P. Mathur, Pearson Education, 2013.
5. Effective Methods for Software Testing, William E. Perry, Wiley, 2nd Edition.
6. Software Testing, Principles and Practices, Srinivasan Desikan, Gopalaswamy Ramesh, Pearson Education, 2006.

Course Code	Course Title				Core/Elective		
U21IT503	DATA STRUCTURES				Open Elective-I		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
-	L	T	D	P	40	60	3
	3	-	-	-			
<b>Course Objectives</b> Develop ability to <ol style="list-style-type: none"> <li>1. Develop skill to design and analyze simple linear and nonlinear data structures, such as stacks, queues and Lists and their applications.</li> <li>2. Gain programming skills to implement sorting and searching algorithms</li> <li>3. Strengthen the ability to identify and apply the suitable data structures for the given real world problem.</li> <li>4. Gain knowledge in practical applications of data structures</li> <li>5. Understand essential for future programming and software engineering courses.</li> </ol> <b>Course Outcomes</b> At the end of the course, student would be able to <ol style="list-style-type: none"> <li>1. Implement various data structures using arrays, linked lists</li> <li>2. Develop ADT necessary for solving problems based on Stacks and Queues</li> <li>3. Implement binary trees, general tree structures, advanced search trees, heaps, graphs.</li> <li>4. Implement hash functions and handle collisions.</li> <li>5. Implement various kinds of sorting techniques and apply appropriate techniques for solving a given problem.</li> </ol>							

**UNIT – I**

**Introduction Data Structures and Algorithms:** Introduction to data structures, classification of data structures, operations on data structures; Algorithm Specification, Recursive algorithms, Data Abstraction.

**Performance analysis-** Time Complexity and Space Complexity, Asymptotic Notation-Big O, Omega, and Theta notations.

**UNIT – II**

**Stacks:** Stack ADT, definition and operations, Implementations of stacks using array, applications of stacks, Arithmetic expression conversion and evaluation.

**Queues:** Queue ADT, definition and operations, Implementation of queues using Arrays, applications of linear queue, circular queue.

**UNIT – III**

**Linked Lists:** Introduction, Singly linked list, representation of a linked list in memory, operations on a Singly Linked List, Implementation of Singly Linked List.

**Doubly Linked Lists:** Operations on Doubly Linked List, Implementation of Doubly Linked List, Circular linked list, Implementation of Stack and Queue using linked list.

**UNIT – IV**

**Searching Techniques:** Linear search and Binary Search algorithms.

**Sorting Techniques:** Bubble Sort, Insertion sort, Selection Sort, Merge Sort, and Quick Sort. Comparison among sorting techniques.

**UNIT – V**

**Trees:** Introduction, Binary Trees, Tree Traversals, Threaded Binary Trees, Binary Search Tree, Heap Tree, AVL Tree.

**Graphs:** Graph Abstract Data Type, Representation of Graph, Graph Traversals -DFS and BFS, Introduction to Spanning Tree.

**Suggested Readings:**

1. Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahani, Susan Anderson Freed, Computer Science Press, 2004
2. Data Structures, S. Lipschutz, Tata McGraw Hill Education, 1st Edition, 2008.
3. Classic Data Structures, D. Samanta, PHI Learning, 2nd Edition, 2004.
4. Data Structures and Algorithm Analysis In C, Mark A Weiss, Second Edition (2002), Pearson.

Course Code	Course Title				Core/Elective		
U21IT607	INTRODUCTION TO WEB PROGRAMMING				Open Elective-II		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
-	3	-	-	-	40	60	3
<b>Course Objectives:</b> Develop ability to <ol style="list-style-type: none"> <li>1. Understand the technologies used in Web Programming.</li> <li>2. Understand the importance of CSS in HTML for styling web pages.</li> <li>3. Develop dynamic web pages using Java Script.</li> <li>4. Understand the document structure and schemas and represent data in that format.</li> <li>5. Understand the basic concepts of PHP for dynamic website designing.</li> </ol> <b>Course Outcomes:</b> At the end of the course, student would be able to <ol style="list-style-type: none"> <li>1. Develop static web pages using HTML and CSS.</li> <li>2. Apply and work with various styling techniques.</li> <li>3. Create dynamic pages using java script.</li> <li>4. Develop web content publishing for applications and work web services.</li> <li>5. Analyze dynamic websites using PHP</li> </ol>							

**UNIT I**

**Introduction to HTML:** Origins and Evolution of HTML and XHTML, Basic Syntax, Standard XHTML Document structure, HTML- Basic tags, Form elements and attributes, validation, Frames, HTML 5.0

**UNIT II**

**Cascading Style Sheets:** Introduction, Levels of Style sheets, Font properties, List Properties, CSS selectors, CSS BOX Model, Background Images.

**UNIT III**

**Basics of Java Script:** Java Script-Object, names, literals, operators and expressions- statements and features- events - windows documents - frames - data types - built-in functions- Browser object model - Verifying forms.

**Unit IV**

**XML and Web Services:** The Syntax of XML, XML Document Structure, Document Type Definitions, Name Space, XML Schemas.

**Web services:** UDDI-WSDL-Java web services – Web resources.

**Unit V**

**Introduction to PHP:** Origins and uses, Overview, general syntax, primitive, operations and expressions, control statements. Comparison of PHP and HTML

**Suggested Readings:**

1. Programming with World Wide Web, Robert W. Sebesta, Eighth Edition, Pearson Education, 2008.
2. Web Programming-Building Internet Applications, Chris Bates ,Second Edition, Wiley 2010
3. Java Script-The Complete Reference, Thomas Powell, Fritz Schneider, Third Edition, Mc graw hill, 2006
4. Learning PHP, My SQL, Java Script, CSS, HTML 5 - A step by step guide to creating dynamic websites, Robin Nixon, Third Edition, OREILLEY, 2011.