

**LORDS INSTITUTE OF ENGINEERING AND TECHNOLOGY****(An Autonomous Institution)****DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING****SCHEME OF INSTRUCTIONS & EXAMINATION [LR-24]****(W.e.f Academic Year 2026-27)****III-B.E. V-Semester****(Tentative)**

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	U24MB501	HSMC	Business Economics and Financial Analysis	3	-	-	3	40	60	3	3
2	U24CD504	PCC	Operating Systems	3	-	-	3	40	60	3	3
3	U24CS501	PCC	Computer Network	3	-	-	3	40	60	3	3
4	U24CD501	PCC	Artificial Intelligence	3	-	-	3	40	60	3	3
5	---	PEC	Professional Elective-I	3	-	-	3	40	60	3	3
<b>Practical/ Laboratory Course</b>											
6	U24CD5L1	PCC	Artificial Intelligence Lab	-	-	3	3	25	50	3	1.5
7	U24CS5L1	PCC	Computer Network Lab	-	-	3	3	25	50	3	1.5
<b>Skill Development Course</b>											
8	U24CS5L2	PCC	Competitive Programming	-	-	3	3	25	50	3	1
<b>Internship</b>											
9	U24CS5P1	PROJ	Summer Internship (During vacation period after IV Semester)	-	-	2	2	50	-	-	1
<b>Total</b>				<b>15</b>	<b>-</b>	<b>11</b>	<b>26</b>	<b>325</b>	<b>450</b>	<b>--</b>	<b>20</b>

**L:** Lecture (*Hrs/Wk/Sem*)    **T:** Tutorial (*Hrs/Wk/Sem*)    **P:** Practical    **D:** Drawing (*Hrs/Wk/Sem*)

**CIE:** Continuous Internal Evaluation

**PCC:** Program Core Course

**OEC:** Open Elective Course

**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.

**SEE:** Semester End Examination

**PEC:** Professional Elective Course

**CS:** Computer Science

Course Code	Course Title				Core/Elective		
U24MB501	Business Economics and Financial Analysis				Core		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
Accounting Fundamentals	3	-	-	-	40	60	3

**Course Objectives:**

Student will be able 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 break- even analysis.
- 4.To acquire the in-depth knowledge on Financial Accounting concepts and principles and preparation of final accounts.
- 5.To understand the financial statements through ration analysis and cash flow techniques.

**Course Outcomes:**

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

Course Code	Course Title				Core / Elective		
U24CD504	OPERATING SYSTEMS				Core		
Prerequisite	Contact Hours Per Week				CIE	SEE	Credits
PPS	L	T	D	P	40	60	3
	3	-	-	-			

**Course Objectives:**

1. To understand the services provided by and the design of an operating system.
2. To understand the structure and organization of the file system.
3. To understand what a process is and how processes are synchronized and scheduled.
4. To understand different approaches to memory management.
5. To Understand Virtual machine concepts, calls for managing processes, memory and the file system

**Course Outcomes:**

At the end of the course, the students will be able to:

1. Understand the fundamental concepts and Functions of operating system.
2. Analyze various scheduling algorithms.
3. Understand deadlock, prevention and avoidance algorithms.
4. Compare and contrast various memory management schemes.
1. Understand the functionality of file systems and perform administrative tasks on Linux Servers

**UNIT-I**

**Introduction:** What Operating system do, Defining operating system, Computer System Operation, Storage Structure, Operating system Structure, Operating System Operations, Computer system architecture: Single Processor, Multi-Processor, Multiprogramming, Multitasking, Process, Memory and Storage Managements, Protection and Security.

**System Structures:** System calls, Types of System Calls, System Programs, System boot

**UNIT-II**

**Processes:** Process concepts, Process Scheduling, Operations on Processes, Inter process communication, Communication in Client/Server Systems

**CPU Scheduling:** Basic Concepts, Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, Priority, Round Robin)

**Synchronization:** Background, The Critical-Section Problem, Peterson's Solution, Synchronization, Dining Philosophers problem Monitors. **Deadlocks:** System Model, Deadlock characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

**UNIT-III**

**Memory-Management Strategies:** Background, Swapping, Contiguous Memory Allocation, Paging, Structure of the Page Table, Segmentation.

**Virtual Memory Management:** Background, Demand paging, Copy-on-write, Page Replacement, Allocation of Frames, Thrashing, Memory-Mapped Files, Allocating Kernel Memory.

**Storage Management:** File System, File Concept, Access Methods, Directory Structure, File-System Mounting, File sharing, Protection.

**UNIT-IV**

**Implementing File Systems:** File System-Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery, Log- Structured File Systems, NFS.

**Secondary Storage Structure:** Overview of Mass-Storage Structure, Disk Structure, Disk Attachment, RAID Structure, Stable-Storage Implementation, Tertiary-Storage Structure.

**UNIT-V**

**Protection and Security:** Goals of Protection, Principles of Protection, Domain of protection, Access Matrix, Implementation of Access Matrix, Access control, Revocation of access rights,

**System Security:** The security problem, program Threats, System and System Network Threats, Cryptography as a Security tool, User Authentication, Implementing Security Defences, firewalling to protect Systems and Networks, Case Studies- Linux System

**Suggested Readings:**

1. Modern Operating Systems by Andrew S. Tanenbaum & Herbert Bos , 5<sup>th</sup> Edition 2022
2. Operating System Concepts, Silberchatz, Galvin & Gagne, 10<sup>th</sup> Edition 2023.

Course Code	Course Title				Core/Elective		
U24CS502	COMPUTER NETWORKS				Core		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
PPS	3	-	-	-	40	60	3

**Course Objectives:**

Develop ability to

1. Become familiar with layered communication architectures (OSI and TCP/IP).
2. Familiarize with flow control and error control mechanisms in data link layer
3. Study the design issues in network layer and various routing algorithms
4. Learn sockets programming and how to implement client/server programs.
5. Understand the concepts of reliable data transfer and how TCP implements these concepts

**Course Outcomes:**

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

1. Explain the function of each layer of OSI and trace the flow of information from one Node to another node in the network
2. Familiarize with the Transmission Media, Flow Control and Error Detection and Correction
3. Understand the principles of IP addressing and internet routing
4. Describe the working of various networked applications such as DNS, mail, file transfer and www
5. Implement client-server socket-based networked applications

**UNIT-I**

**Overview:** Data communication, Networks, The Internet, Protocols and Standards, Layered tasks OSI Model and layers in OSI model, TCP/IP protocol suite, Addressing, Transmission Modes.

**Techniques for Bandwidth utilization:** Multiplexing - Frequency division, Wavelength division, Synchronous and Statistical Time division multiplexing,

**UNIT-II**

**Data Link Layer and Medium Access Sub Layer:** Error Detection and Error Correction, Introduction to Error Detection and error correction, Block coding, CRC.

**Data link control:** Framing, Flow Control and Error control, Protocols-Simplest Protocol, Stop-and-Wait Protocol, Stop-and-Wait ARQ, Go-Hack-N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking.

**Multiple Access:** Random Access-Pure ALOHA, Slotted ALOHA, CSMA/CD, CDMA/CA,

**UNIT-III**

**Network Layer-Switching Techniques:** Circuit switching networks, Packet Switching networks Virtual circuit networks. Logical addressing, Inter Networking, IPV4, IPV6, Tunneling.

**Internet Control Protocols:** Address Mapping -ARP, RARP, BOOTP and DHCP.

ICMP, IGMP. Delivery, Forwarding, Unicast Routing protocol,

**UNIT-IV**

**Transport Layer:** Process to Process Delivery, UDP, TCP, Data Traffic, Congestion, Congestion Control, Congestion Control in TCP. Quality of Service, Techniques to improve QoS.

**UNIT-V**

**Socket Programming:** Socket address, Elementary socket system calls, Advanced socket system calls.

**Application Layer:** Domain Name Space (DNS), EMAIL, SNMP, and Multimedia

**Suggested Readings:**

1. Computer Networks”, Andrew S. Tanenbaum, Pearson Education, Sixth Edition, 2021
2. Data Communication and Networking, 4th Edition, Behrouz A. Forouzan, McGrawHill.
3. Data and Computer Communication, 8th Edition, William Stallings, Pearson Prentice Hall India.
4. W. Richard Stevens, Unix Network Programming, Prentice Hall / Pearson Education, 2009.
5. Unix Network Programming”, W. Richard Stevens, Prentice Hall/Pearson Education, 2009.

Course Code	Course Title				Core/Elective		
U24CD501	ARTIFICIAL INTELLIGENCE				Core		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
Python, DS	3	0	0	0	40	60	3

**Course Objectives:**

The course will introduce the students to

1. To understand the various characteristics of Intelligent agents
2. To learn the different search strategies in AI
3. To learn to represent knowledge in solving AI problems
4. To understand the different ways of designing software agents
5. To know about the various applications of AI.

**Course Outcomes:**

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

1. Use appropriate search algorithms for any AI problem
2. Represent a problem using first order and predicate logic
3. Provide the apt agent strategy to solve a given problem
4. Design software agents to solve a problem
5. Design applications for NLP that use Artificial Intelligence.

**Unit-1:**

**Introduction to Artificial Intelligence and Intelligent Agents:** Introduction to Artificial Intelligence, History and Evolution of AI, Intelligent Systems, Characteristics of Intelligent Systems, Applications of AI, Intelligent Agents, Agent Architectures, Rational Agents, PEAS Representation, Environment Types, State Space Representation.

**Unit-2:**

**Problem Solving and Search Techniques:** Uninformed Search Strategies -Breadth First Search, Depth First Search, Uniform Cost Search, Iterative Deepening Search. Informed Search Strategies- Best First Search, Hill Climbing, A\* Algorithm, Beam Search. Adversarial Search - Minimax Algorithm, Alpha-Beta Pruning, Constraint Satisfaction Problems

**Unit-3:**

**Knowledge Representation and Reasoning:** Knowledge Representation, Knowledge-Based Systems, Propositional Logic, Predicate Logic, Inference Mechanisms, Forward Chaining, Backward Chaining, Semantic Networks, Frames, Ontologies, Expert Systems, Uncertainty in Knowledge Representation, Bayesian Reasoning

**Unit-4:**

**Planning and Natural Language Processing:** Introduction to Planning in AI, Planning Languages, State Space Planning, and Automated Planning. Fundamentals of Natural Language Processing, Text Processing, Syntax and Semantics, Information Extraction, Question Answering Systems, Conversational Agents, and Chatbots.

**Unit-5:**

**Generative AI, Ethics and Emerging Trends:** Introduction to Generative AI, Large Language Models (LLMs), Prompt Engineering, Retrieval-Augmented Generation (RAG), Autonomous Systems, Explainable AI (XAI), Responsible AI, AI Ethics and Fairness, Privacy and Security Issues in AI, Applications of Generative AI, and Future Trends in Artificial Intelligence.

**SUGGESTED READINGS:**

1. S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach", Prentice Hall, **Fourth Edition, 2020**
2. M. Tim Jones, —Artificial Intelligence: A Systems Approach (Computer Science) I, Jones and Bartlett Publishers, Inc.; First Edition, 2008

Course Code	Course Title				Core/Elective		
U24CS502	SOFTWARE TESTING				PE-I		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
SE	3	-	-	-	40	60	3

**Course Objectives:**

The objectives of this course are:

1. To understand and learn the basic concepts of Testing.
2. To follow the methodology of White Box Testing.
3. To Obtain knowledge of Integration and System Testing
4. To understand the concepts of Object-Oriented Testing.
5. To gain the knowledge of complex problems

**Course Outcomes**

At the end of the Course, Student will be:

1. Gain the basic knowledge of Testing.
2. Acquire the knowledge of White Box Testing methods
3. Test an application using Functional Testing.
4. Use Object Oriented Testing and Millennium Testing methods
5. Solve software complexities with best practices

**UNIT - I**

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**

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**

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, cost–benefit analysis of testing, monitoring and control, test reporting, test control Specialized testing, Object Oriented Testing.

**UNIT – V**

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, 5th 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, Gopaldaswamy Ramesh, Pearson Education, 2006.

Course Code	Course Title				Core/Elective		
U24CS503	BASICS OF DATA SCIENCE				PE-I		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
Data Structures	3	-	-	-	40	60	3

### Course Objectives

The Objectives of this course are:

1. To identify the scope and essentially of Data warehousing and Data Mining.
2. To develop research interest towards advances in data mining.
3. To analyze the data, data science life cycle,
4. To analyze the data collection and cleaning, exploratory data analysis
5. To analyze the visualization, statistical inference and prediction, and decision-making algorithms for respective applications.

### Course Outcomes

At the end of the Course, the Student will able to:

1. Understand the basic concepts in data science, including real world applications
2. Understand fundamentals of data and Data Mining Principles.
3. To Understand importance of qualitative data, terminologies related to Data Science.
4. Understand and Extract knowledge using data pre-processing concepts in data science.
5. Understand the basics of R Programming environment: R language, R- studio and R packages

### UNIT-I

**Introduction to Data Science:** Data Science Definition - The Data Science Venn Diagram Terminology  
Types of Data: - Flavors of Data- Structured and Unstructured Data-Quantitative versus qualitative Data  
Four Levels of Data- Case study.

### UNIT-II

**Evolutionary of Data Science:** - Overview of Evolutionary (Five Steps)- Explore the Data Data set 1 - Data set 2 – Titanic Communication Data: Communication matter- Identifying effective and ineffective visualizations- graphs and statistics Be- Verbal Communication.

### UNIT III:

**Basics of Data Mining:** Importance of Data Mining, moving toward the Information Age Data Mining and the Evolution of Information Technology, What Is Data Mining, What Kinds of Data Can Be Mined Database Data, Data Warehouses, Transactional Data, Other Kinds of Data, OLTP & Online Analytical Processing (OLAP), Graphs Database.

### UNIT IV:

**Identification Of Data:** Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity: Euclidean, Jaccard's Index & Cosine Similarity  
**Data Pre-processing:** Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization (ETL Operations).

### UNIT V:

**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, Operations: Arithmetic, Logical and Matrix operations, Data frames, functions, Control structures, Debugging and Simulation in R.

**Suggested Readings:**

1. “Principles of Data Science”, Sinan Ozdemir, Packt Publishing Ltd,2016 (Unit 1& 2)
2. "Data mining concepts and techniques third edition.", Han, Jiawei, Micheline Kamber, and Jian Pei. The MorganKaufmann Series in Data Management Systems 5.4 (2011). (Unit 3 &4)
3. “Practical Data Science with R”, Nina Zumel, Manning Publications, 2014
4. “Principles of Data Science: Mathematical techniques and theory to succeed in data- driven industries”, Sinan Ozdemir Packt Publishing Limited (13 December 2016)
5. “Introducing data science: big data, machine learning, and more, using Python tools”, Cielien, Davy, Arno DB Meysman, and Mohamed Ali, Manning Publications Co., 2016.

Course Code	Course Title				Core/Elective		
U24CS504	Compiler Design				PE-I		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
Cryptography	3	-	-	-	40	60	3

**Course Objectives:**

The objectives of this course are:

1. Understand the fundamentals of assemblers, compilers, linkers, and loaders.
2. Design lexical analyzers using regular expressions and finite automata.
3. Apply parsing techniques and syntax-directed translation methods.
4. Generate intermediate code representations for source programs.
5. Analyze run-time memory management and storage allocation techniques.

**Course Outcomes**

At the end of this course, the student will be able to:

1. Explain the phases and components of language translators.
2. Develop lexical analyzers for token recognition.
3. Construct parsers and syntax-directed translation schemes.
4. Generate intermediate code for compiler implementation.
5. Implement memory management and symbol table techniques in language processing systems.

**UNIT –I**

Introduction to Language Translators: Language Translators; Elements of Assembly Language Programming; Design of the Assembler; Assembler Design Criteria; Types of Assemblers; Working of assembler; Introduction to linker and loader; Compilers; Analysis of the source program; The phases of Compiler; Cousins of Compiler; The grouping of phases; Compiler construction tools.

**UNIT-II**

Lexical Analyzer: Introduction to Lexical Analyzer; Input Buffering; Specification of Tokens; Recognition of Tokens; A Language for Specifying Lexical Analyzers; Finite Automata from a Regular Expression; Design of a Lexical Analyzer Generator; Optimization of DFA; LEX and YACC tools

**UNIT-III**

Syntax Analysis and Syntax-Directed Definitions (SDD): The role of the parser; Context-free grammars; Writing a grammar; Topdown parsing; Bottom-up Parsing; LR parsers; Construction of Syntax Trees; Bottom-Up Evaluation of S-Attributed definitions; L-Attributed Definitions; Syntax Directed Definitions and translation schemes; Error Detection & Recovery; Ad-Hoc and Systematic Methods

**UNIT – IV**

Intermediate Code Generation: Different Intermediate Forms; Syntax Directed Translation Mechanisms and Attributed Mechanisms and Attributed Definition

**UNIT – V**

Run Time Memory Management: Source Language Issues; Storage Organization; Storage Allocation Strategies and Access to Non local Names; Parameter Passing; Symbol Tables; Language Facilities for Dynamic Storage Allocation; Dynamic Storage Allocation Techniques

**Suggested Readings:**

1. A.V.Aho, Ravi Sethi, J.D.Ullman, "Compilers – principles; Techniques and Tools", Pearson .
2. Trembley J.P., Sorenson P.G., "The Theory and Practice of Compiler Writing", Mcgraw-Hill
3. Steven S. Muchnick, "Advanced Compiler Design & Implementation", Morgan Kaufmann Publishers.
4. V. Raghavan, "Principles of Compiler Design", McGrawHill

Course Code	Course Title				Core/Elective		
U24CS604	COMPUTER GRAPHICS				PE-I		
Prerequisite	Contact Hours per Week				CIE	SEE	Credits
	L	T	D	P			
PPS	3	-	-	-	40	60	3

**Course Objectives:**

The objectives of this course are

- 1.To introduce the concept of synthetic camera model, programmable pipeline and OpenGL API.
- 2.To study different interaction modes and data structures that store 2-D and 3-D geometric objects.
- 3.To understand different transformations in 2-D and 3-D.
- 4.To study different rasterization and rendering algorithms

**Course Outcomes:**

On Successful completion of this course, student will be able to

1. Describe the steps in graphics programming pipeline.
2. Write interactive graphics applications using OpenGL geometric primitives.
3. Apply affine transformations for viewing and projections.
  1. Create realistic images of 3-d objects that involve lighting shading aspects

**UNIT – I**

Graphics Systems and Models: Graphics system, Images, Physical and Synthetic, Imaging system, Synthetic camera model, Programming interface, Graphics architectures, Programmable pipelines.

Graphics Programming: Programming two-dimensional applications, OpenGL API, Primitives and attributes, Color, Viewing and Control functions.

**UNIT – II**

Input and Interaction: Input devices, Display lists & modeling, Programming event-driven input, Picking, building interactive models, Animating interactive programs, Logic operations.

Geometric Objects: Three-dimensional primitives, Coordinate systems and frames, Frames in OpenGL, Modeling colored cube.

**UNIT-III**

Transformations: Affine transformations, Transformations in homogeneous coordinates, Concatenation of transformations, OpenGL transformation matrices.

Viewing: Classical and Computer views, viewing with a computer, Positioning of camera, Simple projections, Projections in OpenGL, Hidden surface removal, Parallel-projection matrices, Perspective-projection matrices.

**UNIT-IV**

Lighting and Shading: Light sources, The Phong lighting model, Computational vectors, Polygonal shading, Light sources in OpenGL, Specification of matrices in OpenGL, Global illumination.

From Vertices to Frames: Basic implementation strategies, Line-segment clipping, Polygon clipping, Clipping in three dimensions, Rasterization, Anti-aliasing.

**UNIT -V**

Modeling & Hierarchy: Hierarchical models, Trees and traversal, Use of tree data structure, Animation, Graphical objects, Scene graphs, Simple scene graph API, Open Scene graph, Other tree structures.

**Text Books:**

1. Edward Angel, Interactive Computer Graphics: A Top-Down Approach Using OpenGL, Pearson Education, 5th edition, 2009.
2. Francis S Hill Jr., Stephen M Kelley, Computer Graphics using OpenGL, PrenticeHall Inc., 3rd Edition, 2007.
3. Jim X. Chen, Foundations of 3D Graphics Programming using JOGL and Java3D, Springer Verlag, 2006.
4. Hearn Donald, Pauline M Baker, Computer Graphics, 2nd edition, 1995.

Course code	Course Title				Core/Elective		
U24CD5L1	ARTIFICIAL INTELLIGENCE LAB				Core		
Pre-requisites	Contact Hours Per Week				CIE	SEE	Credits
Python,DS	L	T	D	P			
	-	-	-	3	25	50	1.5
<p><b>Course Objectives:</b> The objective of this lab is to get an overview of the various Artificial Intelligence techniques and can able to demonstrate those using python.</p> <ol style="list-style-type: none"> <li>1. Apply intelligent agent concepts and search algorithms to solve AI problems.</li> <li>2. Implement knowledge representation and reasoning techniques for intelligent systems.</li> <li>3. Analyse game-playing, planning, and constraint satisfaction problems using AI methods.</li> <li>4. Utilize NLP techniques for text processing and conversational AI applications.</li> <li>5. Design AI solutions using LLMs, AI agents, and responsible AI.</li> </ol> <p><b>Course Outcomes:</b> After the completion of the course the student can able to:</p> <ol style="list-style-type: none"> <li>1. Able to design and implement AI solutions searching techniques using AI.</li> <li>2. Able to know about facts of querying.</li> <li>3. Be capable of confidently applying tree mechanism using AI with Neural Network</li> <li>4. Be capable of performing experiments in Machine Learning using real-world data.</li> <li>5. Be capable to implement classifiers and Regression algorithm.</li> </ol>							

**List of Experiments:****WEEK 1:**

1. Design PEAS representations for real-world intelligent systems such as Autonomous Taxi, Smart Home, Medical Diagnosis System, and Robot Vacuum Cleaner.

**WEEK 2:**

1. Develop a simple reflex agent for a Vacuum Cleaner World environment.
2. Implement state space representation for Water Jug Problem.

**WEEK 3:**

1. Implement Breadth First Search (BFS) and Depth First Search (DFS) algorithms.
2. Implement Uniform Cost Search and Iterative Deepening Search algorithms

**WEEK 4:**

1. Implement Best First Search and A\* Search algorithms for path finding problems.
2. Implement Hill Climbing and Beam Search algorithms.

**WEEK 5:**

1. Implement Minimax Algorithm with Alpha-Beta Pruning for Tic-Tac-Toe or similar game.
2. Solve Constraint Satisfaction Problems such as N-Queens or Graph Colouring.

**WEEK 6:**

1. Implement Propositional Logic using Python.
2. Develop a Predicate Logic representation for a real-world problem.
3. Implement Forward Chaining and Backward Chaining inference mechanisms.

**WEEK 7:**

1. Construct Semantic Networks and Frame-based knowledge representations.
2. Implement Bayesian Reasoning for decision-making under uncertainty.

**WEEK 8:**

1. Implement Goal-Based Planning for a simple problem-solving task.
2. Develop a State Space Planning model using suitable planning operators.
3. Perform text preprocessing including tokenization, stop-word removal, stemming, and lemmatization.

**WEEK 9:**

1. Implement Information Extraction from textual data.
2. Develop a Question Answering System using NLP techniques.
3. Design and implement a rule-based conversational chatbot.

**WEEK 10:**

1. Build a simple AI Agent for task automation.

**WEEK 11:**

1. Analyse Explainable AI (XAI) techniques for interpreting AI outputs.

**Experiments Beyond Syllabus**

1. Develop applications using Large Language Models (LLMs) through APIs or open-source frameworks.
2. Implement a Retrieval-Augmented Generation (RAG) system using document retrieval and response generation.
3. Case Study on Responsible AI, Fairness, Privacy, and Security in AI Systems.
4. Mini Project: Development of an AI-based application integrating concepts from Intelligent Agents, Search, Knowledge Representation, NLP, or Generative AI.

Course code	Course Title				Core/Elective		
U24CS5L1	COMPUTER NETWORKS LAB				Core		
Pre-requisites	Contact Hours Per Week				CIE	SEE	Credits
OS	L	T	D	P	25	50	1.5
	-	-	-	3			

### Course Objectives

The Objectives of this Course are:

1. Understand the use of client/server architecture in application development.
2. Understand and use elementary socket system calls in TCP & UDP based sockets.
3. Implement and use advanced socket system calls and TCP and UDP based sockets.
4. Work with the use of concurrent server services.
5. Simulate chat applications using TCP & UDP.

### Course Outcomes

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

1. Understand the usage of basic commands ipconfig, ifconfig, netstat, ping, arp, telnet, ftp, finger, trace route, who is of LINUX platform.
2. Develop and Implement Client-Server Socket based programs using TCP, and UDP sockets.
3. Make a client server communication through TCP and UDP protocols.
4. Expose on advanced socket programming in LINUX environment.
5. Understand transport layer protocols, connection oriented & connectionless models.

Sr No	Experiment No.	NAME OF THE EXPERIMENT
01	Exp-01	<b>Network Devices and Networking Commands basics</b>
02	Exp-02	<b>Using Cisco Packet Tracer (CPT)</b> <b>Build: The Network</b> , Design the topologies, connect 4 computers in the LAN using Switch, Configure IP addressing on all PC using 192.168.1.0/24 network, Check connectivity between all the PC's using Ping command, traceroute, Configure IP Address as diagram & rules, Verify the interface status
3	Exp-03	<b>Configuring Static Routing using Packet Tracer</b> a) Assign IP address according to diagram b) Make sure that interfaces used should be in UP UP state c) Configure static routing d) Verify routing table and reachability between LAN's (Using PING & TRACE commands)
04	Exp-04	<b>Configuring Default Routing using Packet Tracer</b> a) Design the topology b) Assign IP address according to diagram c) Make sure that interfaces used should be in UP-UP state d) Configure Default route used on R1 & R3, Static routing on R2 e) Verify routing table and reachability between LAN's (Using PING & TRACE commands)
05	Exp-05	<b>Configuring Dynamic Routing using Packet Tracer</b> a) Design the topology b) Assign IP address according to diagram c) Make sure that interfaces used should be in UP UP state

		d) Configure Dynamic Routing using EIGRP 100 e) Verify routing table and reachability between LAN's (Using PING & TRACE commands)
06	Exp-06	Write a Program to demonstrate connectionless (UDP) a) Echo Service Using Well Known / Standard Port. b) Write a Program to demonstrate connectionless (UDP) Date and Time Service Using Well Known / Standard Port.
07	Exp-07	a. Write a Program to Demonstrate Connection oriented (TCP) Echo Service Using Well Known / Standard Port b. Write a Program to Demonstrate Connection oriented (TCP) Date and Time Service Using Well Known / Standard Port
08	Exp-08	1. Write a program to implement echo service using RPC. 2. Write a program to implement date and time service using RPC. Program name:day.x 3. Write a program to implement addition of two numbers using RPC.
09	Exp-09	Write a Program to Implement Mini DNS Service
10	Exp-10	Demonstrate by witing a program on Bit Stuffing/ Character stuffing, Link state routing and Distance Vector routing algorithms
11	Exp-Extra	Beyond the syllabus: NS2 simulations for Network analysis

**Suggested Readings:**

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

Course code	Course Title				Core/Elective		
U24CS5L2	COMPETITIVE PROGRAMMING				Core		
Pre-requisites	Contact Hours Per Week				CIE	SEE	Credits
OS	L	T	D	P			
	-	-	-	3	25	50	1

**Course Objectives**

The Objectives of this Course are:

1. To develop problem-solving skills using competitive programming techniques.
2. To understand algorithmic approaches for arrays, strings, and linked lists.
3. To apply data structures such as stacks, queues, and graphs in solving problems.
4. To learn optimization and searching techniques for efficient programming.
5. To enhance logical thinking and coding proficiency for competitive programming contests.

**Course Outcomes**

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

1. Solve computational problems using appropriate algorithms and techniques.
2. Apply data structures to design efficient solutions.
3. Implement graph, string, and recursion-based algorithms.
4. Analyze the time and space complexity of programming solutions.
5. Develop optimized programs for competitive coding and technical interviews.

**COMPETITIVE PROGRAMMING**

1. Majority Element
2. Stock Buy & Sell
3. Kadane's Algorithm
4. Word search
5. Find Duplicate
6. Largest Rectangle in a Histogram
7. Reverse Words in String
8. KMP Algorithm
9. Peak Index in Mountain Array
10. N Queens
11. Knights Tour
12. Middle of a Linked List
13. Rotate a Linked List
14. Implement Stack using Queue
15. Celebrity Problem
16. Circular Tour / Gas Station
17. BFS : Breadth First Search
18. Cheapest Flights within K Stops
19. Assign Cookies
20. Power Set

**Suggested Readings:**

1. Guide to Competitive Programming, Antti Laaksonen, Springer, 2nd edition, 2020.
2. Competitive Programming 3, Steven Halim, Felix Halim, lulu publishers, 2013.
3. Introduction to Algorithms, Cormen, Leiserson, Rivest, Stein, MIT Press, 4th edition, 2022.
4. The Algorithm Design Manual, Steven S, Skiena, Springer Verlag London Limited, 3rd Edition 2020.

Course code	Course Title				Core/Elective		
U24CS5P1	INTERNSHIP				Core		
Pre-requisites	Contact Hours Per Week				CIE	SEE	Credits
	L	T	D	P			
	-	-	2	-	50	-	1

**Course Objectives:**

The objective of the course is to:

1. Produce an accurate record of work performed during the Internship/Co-op
2. Apply engineering knowledge to a problem in industry
3. Produce a technical report
4. Discuss work in a team environment, if relevant to the project
5. Conduct herself/himself responsibly, safely, and ethically in a professional environment

**Course Outcomes:**

At the end of this course, the student will be able to:

1. Design/develop a small and simple product in hardware or software.
2. Complete the task or realize a pre specified target, with limited scope, rather than taking up a complex task and leave it.
3. Learn to find alternate viable solutions for a given problem.
4. Learn to evaluate these alternatives with reference to pre specified criteria.
5. Implement the selected solution and document the same.

**Guidelines:**

Internship is introduced as part of the curricula of 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 Organizations/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 III-year Course. One faculty coordinator will also be attached to the group of 3 students to monitor the progress and to interact with the industry co- ordinate (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

Planning, designing, and reviewing the planned work

5. Executing the plans
6. Documenting progress, experiments, and other technical documentation
7. Further team discussions to discuss results
8. Final report writing and presentation

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

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

**Note:** Students have to undergo summer internship of 4 weeks at the end of semester IV