COURSE CURRICULUM R-19
CSE (4 Year B.Tech Programme) - Total Credits:160

I Year Course structure												
Semester - I												
Course Code	**Title of the course**	**Category**	**Periods**						**Sessional Marks**	**Semester end Exam marks**	**Total Marks**	**Credits**
			L	**T**	**P**	**E**	**O**	**Total**				
CSE111	Engineering Mathematics – I	BS	3	0	0	1	6	10	40	60	100	3
CSE112	Communicative English	HS	3	0	0	1	3	7	40	60	100	3
CSE113	Basic Electronics Engineering	ES	3	0	0	1	3	7	40	60	100	3
CSE114	Digital Logic Design	ES	3	0	0	1	3	7	40	60	100	3
CSE115	Problem Solving With C	ES	3	0	0	1	3	7	40	60	100	3
CSE116	English Language Lab	HS	0	0	3	0	1	4	50	50	100	1.5
CSE117	Problem solving with C – lab.	ES	0	0	3	0	3	6	50	50	100	1.5
CSE118	Environmental Science (Mandatory non-credit course)	BS	3	0	0	0	1	4	50	-	50	-
Total			**18**	**0**	**6**	**5**	**23**	**52**	**350**	**400**	**750**	**18**

Semester - II												
Course Code	**Title of the course**	**Category**	**Periods**						**Sessional Marks**	**Semester end Exam marks**	**Total Marks**	**Credits**
			L	**T**	**P**	**E**	**O**	**Total**				
CSE121	Engineering Mathematics – II	BS	3	0	0	1	6	10	40	60	100	3
CSE122	Engineering Physics	BS	3	0	0	1	4	8	40	60	100	3
CSE123	Engineering Chemistry	BS	3	0	0	1	4	8	40	60	100	3
CSE124	Elements Of Electrical Engineering	ES	3	0	0	1	4	8	40	60	100	3
CSE125	Engineering Drawing	ES	2	0	3	1	4	10	40	60	100	3.5
CSE126	Engineering Physics Lab.	BS	0	0	3	0	1	4	50	50	100	1.5
CSE127	Engineering Chemistry Lab.	BS	0	0	3	0	1	4	50	50	100	1.5
CSE128	Engineering Workshop	ES	0	0	3	0	1	4	50	50	100	1.5
CSE129	Human Values and Professional Ethics(Mandatory non-credit course)	HS	3	0	0		1	4	50	-	50	-
Total			**17**	**0**	**12**	**5**	**26**	**60**	**400**	**450**	**850**	**20**

II Year Course structure

Semester - I

Course Code	SUBJECT NAME	Category	Periods						Sessional Marks	Semester end Exam marks	Total Marks	Credits
			L	T	P	E	O	Total				
CSE 211	DATA STRUCTURES&ALGORITHMS	PC	2	1	0	1	4	8	40	60	100	3
CSE 212	COMPUTER ORGANIZATION	PC	3	0	0	1	4	8	40	60	100	3
CSE 213	JAVA PROGRAMMING	PC	3	0	0	1	4	8	40	60	100	3
CSE 214	DATA COMMUNICATION	PC	3	0	0	1	4	8	40	60	100	3
CSE 215	DISCRETE MATHEMATICAL STRUCTURES	BS	3	0	0	1	4	8	40	60	100	3
CSE 216	DESIGN THINKING & PRODUCT INNOVATION	ES	2	0	2	1	3	8	40	60	100	3
CSE 217	JAVA PROGRAMMING LAB	PC	0	0	3	0	2	5	50	50	100	1.5
CSE 218	DATA STRUCTURES LAB USING C	PC	0	0	3	0	2	5	50	50	100	1.5
Total			**16**	**1**	**8**	**6**	**27**	**58**	**340**	**460**	**800**	**21**

Semester - II

CODE	SUBJECT NAME	Category	Periods						Sessional Marks	Semester end Exam marks	Total Marks	Credits
			L	T	P	E	O	Total				
CSE 221	PROBABILITY , STATISTICS AND QUEUING THEORY	BS	3	0	0	1	6	10	40	60	100	3
CSE 222	MICROPROCESSOR & INTERFACING	PC	2	1	0	2	4	9	40	60	100	3
CSE 223	OPERATING SYSTEMS	PC	3	0	0	1	4	8	40	60	100	3
CSE 224	COMPUTER NETWORKS	PC	3	0	0	1	4	8	40	60	100	3
CSE 225	COMPUTER GRAPHICS	PC	2	1	0	1	4	8	40	60	100	3
CSE 226	FORMAL LANGUAGES AND AUTOMETA THEORY	PC	2	1	0	1	2	6	40	60	100	3
CSE 227	MICRO PROCESSOR INTERFACING LAB	*PC*	0	0	3	0	1	4	50	50	100	1.5
CSE 228	OPERATING SYSTEM LAB	*PC*	0	0	3	0	1	4	50	50	100	1.5
Total			**15**	**3**	**6**	**7**	**26**	**57**	**340**	**460**	**800**	**21**

III Year Course structure

Semester - I

CODE	SUBJECT NAME	Category	Periods						Sessional Marks	Semester end Exam marks	Total Marks	Credits
			L	T	P	E	O	Total				
CSE 311	OPEN ELECTIVE-I*	OE	3	0	0	1	2	6	40	60	100	3
CSE 312	PROFESSIONAL ELECTIVE -I	PE	3	0	0	1	2	6	40	60	100	3
CSE 313	COMPETITIVE PROGRAMMING	PC	2	1	0	1	5	9	40	60	100	3
CSE 314	COMPILER DESIGN	PC	2	1	0	1	4	8	40	60	100	3
CSE 315	DATA BASE MANAGEMENT SYSTEMS	PC	3	0	0	1	4	8	40	60	100	3
CSE 316	DESIGN & ANALYSIS OF ALGORITHMS	PC	2	1	0	1	4	8	40	60	100	3
CSE 317	QUANTITATIVE &VERBAL APTITUDE-I	HS	0	0	3	1	3	7	100	0	100	1.5
CSE 318	DATA BASE MANAGEMENT SYSTEMS LAB	PC	0	0	3	0	1	4	50	50	100	1.5
CSE 319	COMPETITIVE PROGRAMMING LAB	PC	0	0	3	0	1	4	50	50	100	1.5
Total			**15**	**3**	**9**	**7**	**26**	**60**	**390**	**410**	**800**	**22.5**

Semester - II

CODE	SUBJECT NAME	Category	Periods						Sessionals Marks	Semester end Exam marks	Total Marks	Credits
			L	T	P	E	O	Total				
CSE 321	OPEN ELECTIVE -II*	OE	3	0	0	1	2	6	40	60	100	3
CSE 322	PROFESSIONAL ELECTIVE -II	PE	3	0	0	1	2	6	40	60	100	3
CSE 323	PROFESSIONAL ELECTIVE -III	PE	3	0	0	1	2	6	40	60	100	3
CSE 324	OBJECT ORIENTED SOFTWARE ENGINEERING	PC	3	0	0	1	4	8	40	60	100	3
CSE 325	WEB TECHNOLOGIES	PC	2	1	0	1	4	8	40	60	100	3
CSE 326	CRYPTOGRAPHY AND NETWORK SECURITY	PC	3	0	0	1	4	8	40	60	100	3
CSE 327	Quantitative Aptitude-II& SoftSkills	HS	0	0	3	2	3	8	100	0	100	1.5
CSE 328	WEB TECHNOLOGIES LAB	PC	0	0	3	0	1	4	50	50	100	1.5
CSE 329	OBJECT ORIENTED SOFTWARE ENGINEERING LAB	PC	0	0	3	0	1	4	50	50	100	1.5
Total			**17**	**1**	**9**	**8**	**23**	**58**	**390**	**410**	**800**	**22.5**

*Open Elective can be Inter Department Disciplinary Course, Emerging Courses or MOOC. Final decision will be taken by the department.

IV Year Course structure

Semester - I

CODE	SUBJECT NAME	Category	Periods						Sessionals Marks	Semester end Exam marks	Total Marks	Credits
			L	T	P	E	O	Total				
CSE 411	OPEN ELECTIVE -III*	OE	3	0	0	1	2	6	40	60	100	3
CSE 412	PROFESSIONAL ELECTIVE -IV	PE	3	0	0	1	2	6	40	60	100	3
CSE 413	PROFESSIONAL ELECTIVE -V	PE	3	0	0	1	2	6	40	60	100	3
CSE 414	MANAGEMENT PRINCIPLES	HS	3	0	0	0	2	5	40	60	100	3
CSE 415	DATA ANALYTICS	PC	2	1	0	1	4	8	40	60	100	3
CSE 416	CRYPTOGRAPHY & NETWORK SECURITY LAB	PC	0	0	3	0	1	4	50	50	100	1.5
CSE 417	DATA ANALYTICS LAB	PC	0	0	3	0	1	4	50	50	100	1.5
CSE 418	PROJECT -I	PR	0	0	3	0	3	6	100	0	100	2
CSE 419	SUMMER INTERNSHIP-INDUSTRY	PR	0	0	0	0	1	1	100	0	100	1
Total			**14**	**1**	**9**	**4**	**18**	**46**	**500**	**400**	**900**	**21**

Semester - II

CODE	SUBJECT NAME	Category	Periods						Sessionals Marks	Semester end Exam marks	Total Marks	Credits
			L	T	P	E	O	Tot al				
CSE 421	OPEN ELECTIVE -IV*	OE	3	0	0	1	3	7	40	60	100	3
CSE 422	PROFESSIONAL ELECTIVE -VI/MOOC	PE	3	0	0	1	3	7	40	60	100	3
CSE 423	PROJECT -II	PR	0	0	9	0	9	18	100	100	200	8
Total			**6**	**0**	**9**	**2**	**15**	**32**	**180**	**220**	**400**	**14**

*Open Elective canbe Inter Department DisciplinaryCourse,EmergingCourses orMOOC.
Finaldecision willbe taken by the department.

Total Credits 160

		PE1	PE2	PE3	PE4	PE5	PE6
Track 1	Programming and Application Development	Mobile Application Development	Full Stack Web Development	Cloud Application Development and Deployment	Microservices	Principles of Programming Languages	User Experience Design
Track 2	Artificial Intelligence	Artificial Intelligence	Machine Learning & Deep Learning	Data warehousing and Data mining	Intelligent Agents	Bussiness Intelligence	Social Network Analysis
Track 3	Computer Networks and Engineering	Parallel Computing	Embedded Systems	Mobile and Cellular Networks	Embedded Control Systems Design	Microprocessor Design	IOT
Track 4	Systems Engineering	Parallel and Distributed Systems	Virtual Machine essentials	Real-time Systems	Cyber Security and Digital Forensics essentials	Block chain Fundamentals	Edge Computing
Track 5	Virtual and Augmented Reality	Image Processing	Computer Graphics and 3D Design & Printing	Multimedia & Animation	Human Computer Interaction	Virtual, Augmented and Mixed Reality	Gamification
Track 6	Data Processing	Data warehousing and Data mining	Information Modelling and Database Design	Advanced Databases (Parallel, Multimedia, Distributed)	(Left blank for future choice)		

DATA STRUCTURES&ALGORITHMS	
CSE 211	**Credits : 3**
Instruction **:** 3 Periods & 1 Tut/Week	Sessional Marks : 40
End Exam : 3 Hours	End Exam Marks : 60

Prerequisites:

Basic Knowledge of Programming Fundamentals

Knowledge of Programming Languages (C)

Course Objectives:

The course should enable the students:

- To acquire knowledge on several linear and nonlinear data structures like stacks, queues, linked list, trees and graphs.
- To have better insight into to learn various sorting and searching techniques.
- To exercise the applications of data structures.
- To have a good understanding of problem solving using data structure tools and techniques.

Course Outcomes:

the end of the course, the student will be able to:	
1.	Analyze the complexities of recursive and Non recursive algorithms and Implement linear, binary, interpolation, hashing searching techniques and sorting techniques namely bubble, insertion, selection, quick, merge sort.
2.	Apply ADT concepts such as stacks and queues for solving infix to post fix, postfix evaluation, priority queues.
3.	Apply the concepts of dynamic memory allocation for reducing the time and space complexity of algorithms.
4.	Design and implement the Nonlinear data structures (trees) to optimize the solution.
5.	Design and Implement Warshall‟s Algorithm, Shortest path Algorithm-Dijkstra‟s Algorithm, Minimum cost spanning trees (Prims and Krushkals algorithms),Graph traversals (Breadth first search and Depth first Search algorithms.)

Mapping of Course Outcomes with Program Outcomes:

S.No	PO1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO 11	PO12	PSO1	PSO2
CO 1	3	3	2	3	1	1	0	1	1	3	0	2	3	0
CO 2	2	2	3	2	0	0	0	1	1	2	0	2	2	0
CO 3	2	2	3	2	0	0	0	1	1	2	0	2	3	0
CO 4	2	3	3	2	0	0	0	1	1	2	0	3	2	0
CO5	2	3	3	3	0	0	0	1	1	2	0	3	3	0

SYLLABUS

UNIT-I:

15 periods

Introduction: Basic Terminology, Elementary Data Organization, Data Structure operations, Fundamentals of analysis of algorithms and efficiency – Asymptotic Notations and Basic Efficiency classes.

Arrays: Array Definition, Representation and Analysis, Single and Multidimensional Arrays, address calculation, application of arrays, Character String in C, Character string operation, Array as Parameters, Sparse Matrices.

Searching &Sorting: Sequential search, binary search, Interpolation Search, comparison and analysis, Hash Table, Hash Functions. Complexity of Search Algorithm, Insertion Sort, Bubble Sort, Selection Sort, Quick Sort, Merge Sort.

CO1: Analyze the complexity of Algorithms, Implement searching and soring algorithms.
CO2: Implement the searching and soring algorithms.

UNIT-II:

12 periods

Stacks: Array Representation and Implementation of stack, Operations on Stacks: Push & Pop, Applications of stack: Conversion of Infix to prefix and Postfix Expressions, Evaluation of Postfix & Prefix expressions using stack, Recursion, Towers of Hanoi Problem.

Queues: Array representation and implementation of queues, Operations on Queue: Insert, Delete, Full and Empty. Circular queue, De-queue, and Priority Queue, Applications of Queues.

CO1: Implement stacks and queues using ADT and Implement the applications of Stacks and queues (solving infix to post fix, postfix evaluation, priority queues.)

CO2: Apply ADT and implement Stack and queue and applications of stack and queue.

UNIT-III:

12 periods

Linked list: Representation and Implementation of Singly Linked Lists, Traversing and Searching of Linked List, Insertion and deletion to/from Linked Lists, Doubly linked list, Circular doubly linked list, Implementing priority queue using Linked List, Polynomial Representation using Linked list & addition.

CO1: Implement singly linked list, Doubly Linked List, Circular doubly linked list and applications.

CO2: Implement Linked Lists and applications of Linked Lists.

UNIT-IV:

12 periods

Trees: Basic terminology, Binary Trees, Binary tree representation, Almost Complete Binary Tree, Complete Binary Tree, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees. Binary Search Tree (BST), Insertion and Deletion in BST, AVL Trees-Rotations in AVL trees, Insertion and Deletion in AVL.

CO1: Design and implement BST, AVL trees.

CO2: Implement BST, AVL tree along with various operations performed on BST and AVL tree.

UNIT-V:

12 periods

Graphs: Terminology & Representations- Graphs, Directed Graphs, Adjacency Matrices, Path OR Transitive Closure of a Graph, Warshall‟s Algorithm, Shortest path Algorithm-Dijkstra‟s Algorithm, Connected Component and Spanning Trees, Minimum Cost Spanning Trees, Graph Traversals.

CO1: Implement Graph Traversals algorithm and Minimum Cost Spanning Trees algorithms.

CO2: Implemnet Warshall‟s Algorithm, Shortest path Algorithm-Dijkstra‟s Algorithm and Minimum Cost Spanning Trees algorithm

TEXT BOOKS

1. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Pearson Education, 2nd Edition, 1996

REFERENCE BOOKS

1. E.Horowitz and Sahani, "Fundamentals of Data Structures", W H Freeman & Co Publication, 1983.
2. S. Lipschutz, –Data Structures‖, McGraw Hill Publications, 1986.
3. P. Dey & M. Ghosh, –Programming in C‖ , Oxford Univ. Press, 2012
4. ISRD Group, –Data Structures through C++‖, McGraw Hill, 2011.

Web Resources:

1. https://nptel.ac.in/courses/106/102/106102064/
2. https://www.edx.org/course/foundations-of-data-structures
3. https://www.pluralsight.com/courses/ads-part1?aid=7010a000002BWq6AAG&promo=&utm_source=non_branded&utm_medium=digital_paid_search_google&utm_campaign=IN_Dynamic&utm_content=&gclid=CjwKCAjwh472BRAGEiwAvHVfGkBPYPTtMyZXFPv0dFT447PrKa_n8BKqox2DhR-zBq7s4EvOubgD0hoCUyIQAvD_BwE

Prepared By: N. Lokeswari, K. Amaravathi

COMPUTER ORGANIZATION	
CSE 212	**Credits : 3**
Instruction **:** 3 Periods & 3Extended Tut/Week	Sessional Marks : 40
End Exam : 3 Hours	End Exam Marks : 60

Prerequisites:

- Digital Logic Design

Course Objectives:

- To understand the basics of computer hardware and how software interacts with computer hardware.
- To understand the structure, function and characteristics of computer systems.
- To understand the basic structure and operation of digital computer.
- To study the design of arithmetic and logic unit.
- To study the two types of control unit techniques and the concept of pipelining.
- To understand the hierarchical memory system including cache memories and virtual memory.
- To understand the different ways of communicating with I/O devices and standard I/O interfaces.

Course Outcomes:

By the end of the course, the student will be able to:	
1.	Identify the basic principles and apply to arithmetic for ALU implementation.
2.	Examine the functional aspects of processor unit.
3.	Compare and assess the working principles of hardwired and microprogrammed control
4.	Inspect addressing modes, instruction formats in various CPU organizations and Assess the performance implications of processing techniques.
5.	Infer the design issues in memory and I/O organizations.

Mapping of Course Outcomes with Program Outcomes:

Mapping		PO												PSO	
		1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO	1	3	2	2	1					1	1		1	1	1
	2	2	2	2	1									1	2
	3	1	3	2	3									1	1
	4	2	2	2	2									1	2
	5	2	3	3	3					1	1		1	2	2

SYLLABUS

UNIT-1 **12Hours**

Register Transfer and Micro operations :

Register Transfer Language, Bus and Memory Transfers, Arithmetic, Logic and Shift Micro operations, Arithmetic Logic Shift Unit,

Computer Arithmetic:

Introduction, Addition and Subtraction, Booth Multiplication, Division & Decimal Arithmetic Unit Hardware Implementation & Algorithms.

Learning Outcomes:

1. Identify the basic principles of a computer & its Memory Transfers.
2. Apply Arithmetic operations for ALU Implementation.

UNIT-2 **10 Hours**

Basic Computer Organization : Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input-Output and Interrupt, Complete Computer Description.

Learning Outcomes:

1. Analyze the computer Instruction, Instruction codes, Instruction Cycle.
2. Examine the procedure of an Instruction Cycle.

UNIT-3 **9Hours**

Control Design:

Hardwired & Micro Programmed (Control Unit), Control Memory, Address Sequencing, Conditional and Unconditional Branching, Micro program Example.

Learning Outcomes:

1. Specify the design of a control unit in a computer.
2. Distinguish between Hardwired & Micro programmed control unit.

UNIT-4 **12Hours**

Central Processing Unit:

Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes with numerical examples, Data Transfer and Manipulation, Program Control, Program Interrupt, Types of interrupts, CISC Characteristics, RISC Characteristics.

Introduction to Parallel Processing, Pipelining – General Considerations.

Learning Outcomes:

1. Examine addressing modes, Instruction formats in various CPUOrganizations.
2. Analyze the Data processing operations of CPU.

UNIT-5 **17Hours**

Input-Output Organization:

Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, Direct Memory Access.

Memory Organization:

Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory.

Learning Outcomes:

1. Deduce the design issues of Input-output organization.
2. Conclude the Design issues in memory organization of acomputer.

TEXT BOOK

1. M.Morris Mano, –Computer System Architecture|, Pearson Education Inc., 2003, Third Edition.

REFERENCE BOOKS

1. William Stallings, Computer Organization and Architecture, 6th Edition, Pearson/PHI, 2007.
2. Andrew S. Tanenbaum, Structured Computer Organization, 5th Edition, PHI/Pearson, 2007.

ONLINE WEB RESOURCES

1. https://nptel.ac.in/courses/106/103/106103068/

2. https://freevideolectures.com/course/2277/computer-organization

---

Prepared by: T.Anitha, G.Gayatri.

CSE 213 **Object oriented Programming with JAVA**

CREDITS: 3

INSTRUCTION: 3Theory & 1Tutorial/ Week SESSIONAL MARKS: 40

FINAL EXAM: 3Hrs FINAL EXAM MARKS: 60

Prerequisites:

- Basic knowledge of computer fundamentals.
- Student must have knowledge of some programming languages (such as C, C++)

Objectives:

- To Understand Object Oriented Programming Concepts and Apply Them in Problem Solving.
- To Learn The Basics of Java Console and GUI Based Programming.

Course Outcomes:

CO-1: Design Classes for Real Time Applications.
CO-2: Establish The Connectivity Among The Classes Using Inheritances And Interfaces.
CO-3: Modularize The Application Using Packages and apply threads on classes to achieve parallelism through synchronization.
CO-4: Develop Test Cases By Including The Runtime Errors Using Exceptions Handling Mechanism and multi Threading
CO-5: Identify AWT components to Design the GUI Using Applet & AWT Frameworks

CO-PO MAPPING:

	PO- 1	**PO- 2**	**PO- 3**	**PO- 4**	**PO- 5**	**PO- 6**	**PO- 7**	**PO- 8**	**PO- 9**	**PO-10**	**PO-11**	**PO-12**	**PSO-1**	**PSO- 2**
CO-1	1	2	3	1	1	-	-	-	1	-	1	3	2	2
CO-2	2	3	3	2	2	-	-	-	2	-	-	3	2	2
CO-3	1	3	3	1	3	-	-	-	2	-	-	3	2	2
CO-4	1	2	3	2	2	1	-	-	2	-	-	3	2	2
CO-5	2	1	3	2	3	-	-	-	2	-	-	3	2	2

Correlation Levels 1 2 3 Defined as Below

1 High: Strong Correlation
2 Medium: Moderate Correlation
3 Low: Slight

COURSE CONTENTS:

UNIT-I **10-12hours**

Fundamentals of Object Oriented Programming : Introduction, object oriented paradigm, object and classes, Data Abstraction and Encapsulation, Inheritance, Polymorphism, Dynamic Binding, Applications of OOP.

Java programming - History of Java, Java Buzzwords, Data types, variables, operators. Control structures, arrays, console input and output, Simple programs on java. Introduction to Classes, objects, constructors, methods, parameter passing, overloading constructors and methods, static fields and methods, this reference, final keyword, garbage collection, finalize method, inner class, String handling.

Learning Outcomes: At the end of this unit the Students will be able to

1. Identify the object and understand object oriented principles
2. Create class, constructor and can handle string operations

UNIT – II **10-12 hours**

Inheritance – Basics, using super keyword, multilevel hierarchy, Member access rules, preventing inheritance- using final, the Object class and its methods.

Polymorphism - dynamic binding, method overriding, abstract class and methods. Interfaces - Interfaces vs. Abstract class, defining an interface, implementing interfaces, accessing implementations through interface references, extending interfaces.

Packages - Defining, Creating and Accessing a Package, importing packages

Learning Outcomes: At the end of this unit the Students will be able to

1. Derive a class from existing class or from interface
2. Define a package and importing class from package

UNIT –III **10-12 hours**

I/O: I/O basics, byte and character streams, read/ write console input/output, reading and writing files.

Exception handling – Fundamentals, Exception types, use of try and catch, throw, throws, finally, multiple catches, built-in exceptions, user defined exceptions.

Multithreading – Thread Priorities, synchronization, messaging, reading a thread, creating multiple threads, use of alive and join, inter-thread communication- suspending resuming and stopping threads, producer-consumer problem with multithreading.

Learning Outcomes: At the end of this unit the Students will be able to

1. Handle predefined Exceptions and can define custom exceptions
2. Split a complex task into multiple threads.

UNIT-IV **10-12 hours**

Swings: Introduction to swings, The HTML Applet tag, a simple banner applet. Difference between Application program and applet program.

AWT-Working with Windows, Graphics and Text: AWT Classes and components, Window fundamentals, working with Frame windows, Working with graphics,

Layout Managers : Flow Layout, Border Layout, Grid Layout.

Learning Outcomes: At the end of this unit the Students will be able to

1. Design Swing Applet class with html tag
2. Design GUI components using AWT
3. Arrange components in Layouts

UNIT-V **10-12 hours**

Event Handling: The Delegation event model, Event classes, Event Listener interfaces, handling Action event, Item Event, Mouse Event, keyboard event and Window Events.

Enterprise Java Beans: Introduction to EJB, Advantages and Disadvantages of EJB, Difference between RMI and EJB, Types of EJB: Session bean, entity bean, message driven bean.

Learning Outcomes: At the end of this unit the Students will be able to

1. Define Event Handling on the components using Delegation event model
2. Examine different types of beans

TEXT BOOKS

1. Herbert Schildt, ―JAVA The Complete Reference‖, TataMcGraw Hill,seventh edition.
2. ―Programming with JAVA - A Primer‖ – Third Edition, E Balagurusamy.

REFERENCES BOOKS

1. P.J. Deitel and H.M. Deitel, ―Java for Programmers‖, Pearson education (OR) P.J. Deitel and H.M. Deitel, ―Java: How to Program‖, PHI.
2. P. Radha Krishna, ―Object Orientd Programming through Java‖, Universities Press.
3. Bruce Eckel, ―Thinking in Java‖, Pearson Education
4. Bruce Eckel, ―Programming in Java‖, Pearson Education
5. S. Malhotra and S. Choudhary, ―Programming in Java‖, Oxford Univ. Press.

Faculty Names:

1. Mrs. G. Pranitha
2. Mrs. B. Siva Jyothi

CSE 214 **DATA COMMUNICATION** **CREDITS: 3**

L T P E O **SESSIONAL MARKS: 40**
3 0 0 1 0
END EXAM: 3Hrs **FINAL EXAM MARKS: 60**

Pre -requisites:
Basic knowledge of Computer fundamentals, Digital Logic Design, Computer Hardware.

Course Objectives:

1 To educate concepts, vocabulary and techniques currently used in the area of Data Communication, Networking and Internet.
2 To interpret the Digital encoding Techniques in Data Communication.
3 Familiarize the student with the basic taxonomy and terminology of the Data and signals, Signal Transmission, and Transmission Impairments.
4 To accumulate existing state-of-the-art in Data Link Layer concepts and sliding window protocols and its applications.
5 To analyze the functions of physical layer and gain knowledge in different mediums used for data transfer.
6 Introduce the student to illustrate the point in Data Communication & networking concepts, preparing the student for those entry level courses.

Course Outcomes:

By the end of the semester, the student will be able to:
CO-1: Acquire the Basics of data communications model, differentiate OSI with TCP/IP models .
CO-2: Analyse and explain the features of Transmission media with various encoding techniques and examine the transmission impairments.
CO-3: Apply the error correction and detection techniques.
CO-4: Analyse the performance issues of different types of LANs
CO-5: Analyse the characteristics of multiplexing and spread spectrum.

CO – PO Mapping:

	PO-A	PO-B	PO-C	PO-D	PO-E	PO-F	PO-G	PO-H	PO-I	PO-J	PO-K	PO-L	PSO-1	PSO-2
CO-1	2	2	3	2	3				2	1		2	1	1
CO-2	2	2	2	1	1				2	2		1	1	1
CO-3	1	2	3	2	1				3	2		2	1	1
CO-4	3	2	2	1	1				2	1		1	1	1
CO-5	2	1	2	2	1				3	2		2	1	1

Syllabus:

UNIT-I **10-hours**

Introduction to Data Communications:
History of Data communication, Data Communication System: An Example Configuration, Protocol Architectures: Reference Models, ISO-OSI model, The TCP/IP. Traditional Internet-Based Applications, Analog and Digital Data Transmission.

Learning Objectives: At the end of this unit student will be

- Emphasis on what Representations being used for defining data communication system with model.
- Analyze the performance comparison between ISO-OSI model and TCP/IP models.

UNIT-II **12 -hours**

Transmission Media:
Guided Media, Un- Guided Media: Antenna, characterstics, Wireless propagation Techniques.
Representation of Signals: Analog Signal-Digital Signal, Amplitude, Frequency, Phase, Data/Signal Encoding Techniques, Modulation and Demodulation techniques, Data-Analog data, Digital Data, Transmission Impairments.

Learning Objectives: At the end of this unit student will be

- describe the characteristics of guided and unguided media with internal representation.
- Analyze the signals with encoding techniques to present the target data.

UNIT –III **14 -hours**

The Digital Data Communication Techniques:
Data Link Control, Asynchronous and Synchronous Transmission, Line Configurations, interface standards-DCE, DTE, Modems.
Flow Control-Error Control:Stop and Wait, Go Back -N, selective repeat protocols,ARQ protocols.
Error- Error Detection, and correction- Parity Check, VRC, CRC, Hamming code, Bit oriented protocol-HDLC,LAP,ATM-cell relay.

Learning Objectives: At the end of this unit student will be

- Understanding the state and types of digital transmissions.
- Analyze the flow and error control methods .

UNIT-IV **12 -hours**

Local Area Network and IoT:
LAN addressing, LAN Topologies, LAN Technologies, LAN Protocol Architecture, Bridges, Layer 2 and Layer 3 Switches.
High-Speed LANs: Ethernet (IEEE 802.3, IEEE 802.5)-Frame format, Delays, Specifications, Wireless LANs: IEEE 802.11- Architecture and Services.
Short Range networks: Bluetooth- Architecture and Services.
Introduction to IoT : Design principles, Architecture, Protocols, Applications.

Learning Objectives: At the end of this unit student will be

- Define and interpret the LAN architecture and its variants.
- Understanding the view of High speed LAN and its applications.
- Illustrate the technologies Ethernet,IEEE802.11, IoT.

UNIT-V **12 -hours**

Multiplexing: Caterories, Demultiplexing: Categories Frequency-Division Multiplexing: characteristics. Time Division Multiplexing-Characteristics: Types, Statistical TDM, Synchronous TDM: Characteristics, TDM Link Control, The Concept of Spread Spectrum, Frequency Hopping Spread Spectrum, Direct Sequence Spread Spectrum, Code-Division Multiple Access.
Case Study: Analog services and Digital services

Learning Objectives: At the end of this unit student will be

- Compare the multiplexing techniques with respect to Frequency and time.
- Discuss the various spread spectrum techniques.

TEXT BOOKS

1. Behrouz A. Forouzan, –Data Communications and Networking‖, TMH, 2004, 4th Edition.
2. Raj kamal,‖ Internet of Things – Architecture and Design principles―, Publisher: Tata McGraw-Hill Education India.

REFERENCE BOOKS

1. William Stallings, ―Data and Computer Communications‖, Pearson Education Inc., 2010, 8th Edition.

Online Web Resources:

1. https://nptel.ac.in/courses/106108098/#
2. https://www.globalitschools.com/programming-quiz/desktop-programming-quiz/data-communication/

Prepared By: Mr. G.Jagadish**,** Dr. V.Usha Bala

Course Code 215	DISCRETE MATHEMATICAL STRUCTURES [common to CSE & I.T.]	L 2	T 1	P 0	C 3

1. Prerequisites : Elementary knowledge of Set theory, Matrices and Algebra.

2. Course Objective :

The main objectives of the course are to:
- Introduce concepts of mathematical logic for analyzing propositions and proving theorems.
- Use sets for solving applied problems binary relations and introduce concepts of algebraic structures
- Work with an ability to solve problems in Combinatorics
- Solve problems involving recurrence relations and generating functions.
- Introduce basic concepts of graphs, digraphs and trees

3. Course Outcomes: At the end of the course student should be able to:

CO - 1	Understand mathematical logic, mathematical reasoning and to study about the validity of the arguments and also prove mathematical theorems using mathematical induction.
CO - 2	Determine properties of binary relations, identify equivalence and partial order relations, sketch relations and Familiarize with algebraic structures.
CO - 3	Apply counting techniques to solve combinatorial problems and identify, formulate, and solve computational problems in various fields.
CO - 4	Understand Recurrence Relation, Generating functions and solving problems involving recurrence equations.
CO - 5	Familiarize with the applications of graphs , trees and algorithms on minimal spanning tress and apply graph theory in solving computing problems

DETAILED SYLLABUS:

UNIT-I: MATHEMATICAL LOGIC (12Periods)

Fundamentals of logic- Logical inferences-Methods of proof of an implication-First order logic and other proof methods -Rules of inference for quantified propositions – Mathematical induction.

Learning outcome: At the end of this unit, student will be able to
- Find equivalence formulas, implementation of logic for mathematical proofs (L_1)
- Apply inference theory to verify the consistence of data (L_3)
- Construct logical statements from informal language to propositional logic expressions(L_6)
- Apply the pigeonhole principle in the context of a contradiction proof (L_3)
- Prove mathematical theorems using mathematical induction(L_5)

(Sections: 1.5 to 1.10 of Text book [1])

UNIT-II : RELATIONS AND ALGEBRAIC SYSTEMS (12 Periods)

RELATIONS:

Cartesian products of sets –Relations - Properties of binary relations in a set – Relation matrix and graph of a relation – Partition and covering of set – Equivalence relations - Composition of Binary relations-Transitive closure of a relation -Partial ordering – Partially orderedset.

(Sections :2-1.9,2-3.1 to 2-3.5,2-3.7, 2-3.8, 2-3.9 of Text book [2])

ALGEBRAIC SYSTEMS:

Definitions and simple examples on Semi groups, Monoids , Group, Ring and Fields.

Learning Outcomes:

Learning outcome: At the end of this unit, student will be able to

- Determine properties of relations, identify equivalence and partial order relations, sketch relations. (L_5)
- Understand concepts of Semi group, Monoid, Group, Ring and Fields. (L_2)

(Sections:3-1.1, 3-2.1,3-2.2, 3-5.1,3-5.11and 3-5.12 of Text book [2])

UNIT-III: **ELEMENTARY COMBINATORICS (10Periods)**

Basics of counting- Combinations and permutations-Their enumeration with and without repetition-Binomial coefficients-Binomial and multinomial theorems-The principle of inclusion-Exclusion.

Learning outcome: At the end of this unit, student will be able to

- Solve problems on Permutation and Combinations with and without repetition (L_3)
- Solve problems on binomial and Multinomial coefficients(L_3)
- Solve counting problems by using principle of inclusion-exclusion (L_3)

(Sections :2.1to 2.8 of Text book [1])

UNIT-IV: RECURRENCE RELATIONS (10Periods)

Generating functions of sequences-Calculating their coefficients-Recurrence relations-Solving recurrence relations-Method of characteristic roots- Non-homogeneous recurrence relations and their solutions.

Learning outcome: At the end of this unit, student will be able to

- Formulate recurrence relations of the sequences
- Solve problems using generating functions(L_3)
- Solve homogeneous linear recurrence relations(L_3)
- Evaluate complementary function and particular integral for non homogeneous linear recurrence relations (L_5)
- Apply substitution method to solve non-linear recurrence relations (L_3)

(Sections: 3.1 to 3.6 of Text book [1])

UNIT- V: GRAPH THEORY (16Periods)

Introduction to graphs – Types of graphs – Graphs basic terminology and special types of simple graphs - representation of graphs and graph isomorphism – Euler paths and circuits- Hamilton paths and circuits – Planar graphs – Euler‟s formula

Introduction to Trees and their properties – Spanning Trees — Minimum Spanning Trees – Kruskal‟s Algorithm .

Learning outcome: At the end of this unit, the student will be able to

- Identify different graphs and their properties(L_3)
- prove elementary results about graphs and trees(L_5)
- Construct Euler and Hamiltonian graphs (L_3)
- Construct the graph for the given data (L_3)
- Construct the spanning tree and binary tress from graphs (L_3)
- Build minimal spanning tree by using different algorithms (L_3)

(Sections: 5.1 to 5.4, 5.7,5.8,5.9,5.10 of Text book [1])

TEXT BOOKS:

1). Joe L. Mott, Abraham Kandel & T. P. Baker, ‒Discrete Mathematics for computer scientists & Mathematicians‖ Prentice Hall of India Ltd, New Delhi.,2008

2) J.PTremblay,R.Manohar,―DiscreteMathematicalStructures with Applications to Computer Science‖, Tata McGraw-Hill Publishing Company Limited,1997

REFERENCE BOOKS:

1. Keneth. H. Rosen, Discrete Mathematics and its Applications, 6/e, Tata McGraw-Hill, 2009.
2. Richard Johnsonburg, Discrete mathematics, 7/e, Pearson Education, 2008.

ANITS (A)
CSE - DEPARTMENT

II YEAR – I SEMESTER

DESIGN THINKING AND PRODUCT INNOVATION											
Code	**Category**	**Periods**						**Sessional Marks**	**End Exam Marks**	**Total Marks**	**Credits**
		L	**T**	**P**	**E**	**O**	**Total**				
CSE 216	**ES**	**2**	**0**	**2**	**1**	**3**	**8**	**40**	**60**	**100**	**3**

Prerequisite: NIL

Course Objectives:

The course titled Innovation, Business Models and Entrepreneurship is designed to give an in-depth Understanding on Various aspects of Innovation, Creativity, evolving business models, incubation and entrepreneurship. Come up with exposure to design thinking for designing innovative products. The course is a blend of theory and practice therefore this course does not require any prerequisite and will be useful to understand innovation and its applications in different spheres of development and growth

Course Outcomes:

The Student will be able to:

CO-1	plain the fundamentals of Design Thinking and innovation
CO-2	pathize and analyse model action plan.
CO-3	scribe the principles of innovation and idea generation for product design
CO-4	ply design thinking techniques for given tasks.
CO-5	ply the design thinking techniques for solving problems in various sectors.

Mapping of Course Outcomes with Programme Outcomes.

High-3, Medium-2, Low-1

COURSE OUTCOMES	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	**3**	**3**	**2**				**1**			**1**	**1**	**1**
CO2	**3**	**3**	**1**				**1**			**1**	**1**	**1**
CO3	**3**	**3**	**2**				**1**			**1**	**1**	**1**
CO4	**3**	**2**	**1**				**1**			**1**	**1**	**1**
CO5	**3**	**3**	**2**				**1**			**1**	**1**	**1**

Course Outcomes	**PSO1**	**PSO2**
CO-1	**2**	**1**
CO-2	**2**	**1**
CO-3	**2**	**1**
CO-4	**2**	**1**
CO-5	**2**	**1**

SYLLABUS

UNIT – I **6L+4P=10 Periods**

Introduction to Design Thinking: Introduction to elements and principles of Design, basics of design-dot, line, shape, form as fundamental design components. Principles of design. Introduction to design thinking, history of Design Thinking, New materials in Industry.

UNIT-II **6L+8P=14Periods**

Design thinking: Design thinking process (empathize, analyze, idea & prototype), implementing the process in driving inventions, design thinking in social innovations. Tools of design thinking - person, costumer, journey map, brain storming, product development.

UNIT – III **6L+4P=10Periods**

Innovation: Art of innovation, Difference between innovation and creativity, role of creativity and innovation in organizations. Creativity to Innovation. Teams for innovation, Measuring the impact and value of creativity.

Product Design: problem formation, introduction to product design, Product strategies, Product value, Product planning, product specifications.

UNIT –IV **6L+6P=12Periods**

Design thinking for strategic Innovation: An exercise in design thinking – implementing design thinking for better process. Implement design thinking process in various Industries. Design thinking for Startups.

UNIT-V **6L+8P=14Periods**

Design thinking in various sectors: Case studies in Information Technology, Finance, Education, Management and Retail sector. Analyze and Prototyping, Usability testing, Organizing and interpreting results.

Case study learning outcomes:

1. Make use of practical design thinking methods in every stage of problem with the help of method templates.
2. Apply design thinking to a problem in order to generate innovative and user-centric solutions.
3. Empathize with end user and initiate a new working culture based on user-centric approach.
4. Prototype and run usability tests for unbiased examination of the product in order to identify problem areas.

Text Books:

1. Change by design, Tim Brown, Harper Bollins (2009)
2. Design Thinking in the Class Room by David Lee, Ulysses press

Reference Books:

1. Design the Future , by Shrrutin N Shetty , Norton Press
2. Universal principles of design- William lidwell, kritina holden, Jill butter.
3. The era of open innovation – chesbrough.H
4. Product Design and Manufacturing by A.K. Chitale and R.C. Gupta, Prentice Hall

Web References:

https://drive.google.com/file/d/1cplqb1eOWnoNMhFWNP8TyYLF2qHdGY_K/view
https://nptel.ac.in/courses/110/106/110106124/#

CSE 217 JAVA Lab CREDITS: 2
INSTRUCTION: 3 Hrs/ Week SESSIONAL MARKS: 50 FINAL
EXAM: 3Hrs FINAL EXAM MARKS: 50

Prerequisites:

- Basic knowledge of computer fundamentals.
- Student must have knowledge of some programming languages (such as C ,C++)

Course Objectives:

- To understand object oriented programming concepts, and apply them in problem solving.
- To learn the basics of java Console, GUI based programming and networking programming.

Course Outcomes:

Student will be able to

- CO-1: Apply basics of Java programming and analyze oops concepts.
- CO-2: Modularize the application using packages.
- CO-3: Apply multi threading and Exception handling.
- CO-4: Design GUI applications using java AWT and applets**.**

CO-PO MAPPING:	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	PSO1	PSO2
CO-1	2	2	1	2	2	0	1	1	2	1	0	2	2	2
CO-2	2	3	3	3	3	1	1	1	3	1	0	2	2	2
CO-3	1	2	3	3	3	1	1	1	3	1	0	2	2	2
CO-4	2	1	3	1	3	1	1	1	3	1	0	2	2	2

JAVA LAB PROGRAMS

1. Write a java program which reads your name and other details through command line and print them.

2. Strings

Write a program to find the longest Substring without Repeating Characters

Input: abcabcbb output:3 string: abc

Input: pwwkew output:3 string: wke Note: pwke is not a substring , it is a subsequence

3. Arrays

Write a program to find the Valid Parentheses

Given a string containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid. An input string is valid if:

1. Open brackets must be closed by the same type of brackets.
2. Open brackets must be closed in the correct order.

Note that an empty string is also considered valid. Input:( ) output:valid

Input: ({) } Output: Not valid

4. Letter Combinations of a Phone Number

Given a string containing digits from `2-9` inclusive, return all possible letter combinations that the number could represent.

A mapping of digit to letters (just like on the telephone buttons) is given below. Note that 1 does not map to

any letters.

Example: Input: "23"

Output: ["ad", "ae", "af", "bd", "be", "bf", "cd", "ce", "cf"].

5. Classes and Objects
Design a –farm animals‖ java application with the details of animals like cow, pig, horse.Consider the following details like where they stay, what they eat,the sound they make by using classes and objects.

6. Constructor overloading
An organization is maintaining the data of employee according to cadre of employee with following parameters name,id,designation,salary,promotion status. Apply the constructor overloading to implement it.

7. Method overriding

All the banks operating in India are controlled by RBI. (e.g. minimum interest rate, minimum balance allowed, maximum withdrawal limit etc) which all banks must follow. For example, suppose RBI has set minimum interest rate applicable to a saving bank account to be 4% annually. however, banks are free to use 4% interest rate or to set any rates above it.

Write a JAVA program to implement bank functionality in the above scenario and demonstrate the dynamic polymorphism concept. Note: Create few classes namely Customer, Account, RBI (Base Class) and few derived classes (SBI, ICICI, PNB etc). Assume and implement required member variables and functions in each class.

Testcase1:
Enter the Bank name to find the rate of Interest : RBI
RBI rate of interest is : 4%

Testcase2:
Enter the Bank name to find the rate of Interest : SBI
RBI rate of interest is : 7%

8. Interfaces:
Different categories of employees are working in a software company like Regular Employees, Contract Employees and Vendors. And their pay roll is different for regular and contract employees. For the regular employees Basic pay is 25000, HRA is 15000rs and TA is 5000. For the Contract employees Basic pay is 12000 TA is 3000rs and there is no HRA. Find the monthly salary details of Employee. If input is Regular Employee display the Regular employee salary details. If input is Contract based display the Contract salary details.
TestCase1: Input: Enter Employee Id: R101
Output: Salary Details:
Basic Pay: 25000 HRA: 15000 T.A: 5000 Total Amount: 45000

9. Packages

Define a package **number** and in that define **Roman class** and implement **romanToInteger()** and import the method in another class.

Input: "LVIII" **Output:** 58

Explanation: L = 50, V= 5, III = 3.

10. File Handling

Write the below text in the file called sample.txt and then find the frequency count of the patterns „pe", and „pi"

Peter Piper picked a peck of pickled peppers
A peck of pickled peppers Peter Piper picked
If Peter Piper picked a peck of pickled peppers
Where"s the peck of pickled peppers Peter Piper picked?

Expected Output:

„pe' – no of occurrences - 20
„pi" – no of occurrences – 12

10. Exception Handling

Input a mobile number and check the given number is valid mobile number or not.

- A valid mobile number is a combination of (0-9) digits of length exactly 10.
- If the given Number Exceeds length of 10 raise Invalid Mobile Number-ArrayIndexOutofBounds Exception
- If the given Number less than the length of 10 raise Invalid Mobile Number – LengthNotSufficientException
- If the given Number contain any character other than digit raise Invalid Mobile Number – NumberFormatException

Sample Input	Expected Output – 1
9885089465	Valid number
98567890121	Invalid Mobile Number-ArrayIndexOutofBounds Exception
88664433	Invalid Mobile Number – LengthNotSufficientException
98ab@123	Invalid Mobile Number –NumberFormatException

11. Multi Threading

Implement a Reservation system which allows the persons to book seats. Define reserve method initially with 100 seats. Now create one or more person threads to book seats. At any time it should allow only one person thread to access the reserve method.

Expected Output:

Person-1 entered.
Available seats: 10 Requested seats: 5
Seat Available. Reserve now :-)
5 seats reserved.
Person-1 leaving.

Person-2 entered.
Available seats: 5 Requested seats: 2
Seat Available. Reserve now :-)
2 seats reserved.
Person-2 leaving.

Person-3 entered.
Available seats: 3 Requested seats: 4
Requested seats not available :-(
Person-3 leaving.
---

12. Design a mini application using the java components.

Faculty Names:

3. Mrs. G. Pranitha
4. Mrs. B. Siva Jyothi

CSE 218 **DATA STRUCTURES** **Credits: 1.5**
Instruction: 3 Periods/week **LAB USING C** Sessional Marks: 50
End. Exam: 3 Hours End-Exam-Marks: 50

Outcomes of the Lab:

1. Implement the techniques for searching and sorting (quick and merge).
2. Implement of stack and queue and Linked list data structures and their applications.
3. Implement operations like insertion, deletion, search and traversing mechanism on binary search tree
4. Apply BFS and DFS algorithms to implement graph traversal.

CO-PO Mapping:

Mapping		PO												PSO	
		1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO	1	2	2	2	1		1		1	1	1		1	1	
	2	1	2	2	1		1		1	1	1		1	1	
	3	2	2	2	1		1		1	1	1		1	2	1
	4	2	2	2	1		1		1	1	1		1	2	

Experiments:

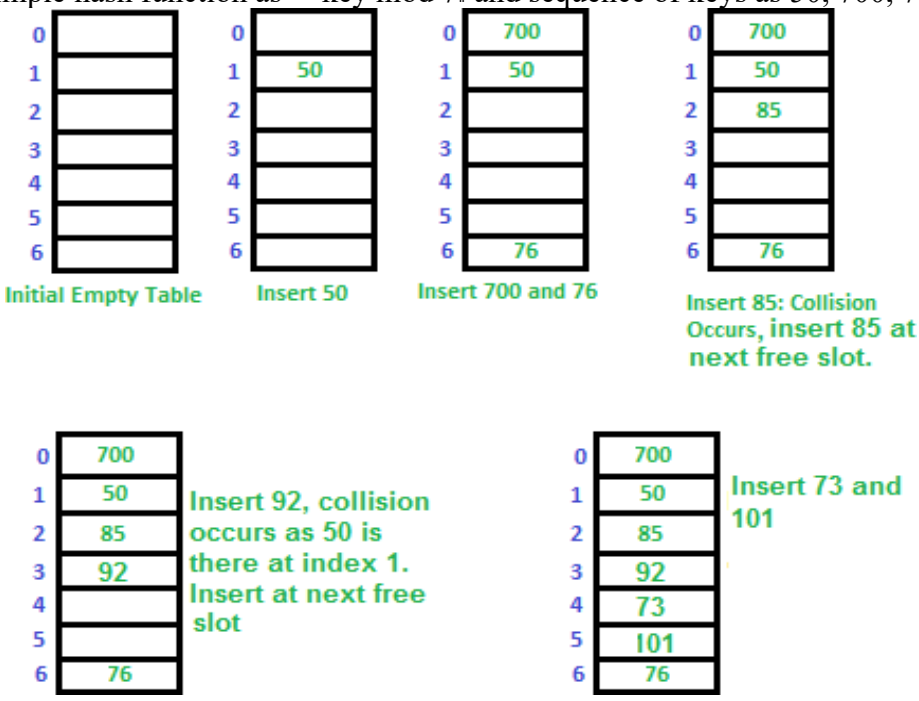
1. Write a program to sort the given array of N elements using divide and conquer method (merge sort and quick sort algorithms) CO1
Constraints: 1<N<1000
Sample Input array: 87, 36, 9, 12, 24, 5, 78, 567, 456, 34, 96, 45, 39, and 89,123
Sample Output array: 5, 9, 12, 24, 34, 36, 39, 45, 78, 87, 89, 96, 123, 456, and 567

2. Write a C Program to search whether an item K present in an array of N elements (Using Linear and binary Search algorithms) CO1
Constraints: 1<K<1000
1<N<1000
Sample Input array: 45, 78,123, 48, 34, 89, 67, 54, and 74,543
Search Item: 34 Search Item: 343
Output: Key Found Output: Key Not Found

3. Write a C program to store k keys into an array of size n at the location computed using a hash function, loc = key % n, where k<=n and k takes values from [1 to m], m>n. CO1
4. Design, Develop and Implement a C program to handle the collisions using the following collision resolution Technique CO1
a) **Linear probing**: In linear probing, we linearly probe for next slot, let store k keys into an array of size S at the location computed using a hash function, hash(x) where k<=n and k takes values from [1 to m], m>n.
Constraints: If slot hash(x) % S is full, then we try (hash(x) + 1) % S
If (hash(x) + 1) % S is also full, then we try (hash(x) + 2) % S
If (hash(x) + 2) % S is also full, then we try (hash(x) + 3) % S
.....
.....

Sample Test Case:

Let us consider a simple hash function as —key mod 7‖ and sequence of keys as 50, 700, 76, 85, 92, 73, 101.



b) **Quadratic probing:** Quadratic Probing we look for i2„th slot in i‟th iteration, let store k keys into an array of size S at the location computed using a hash function, hash(x) where k<=n and k takes values from [1 to m], m>n.

Constraints: let hash(x) be the slot index computed using hash function.

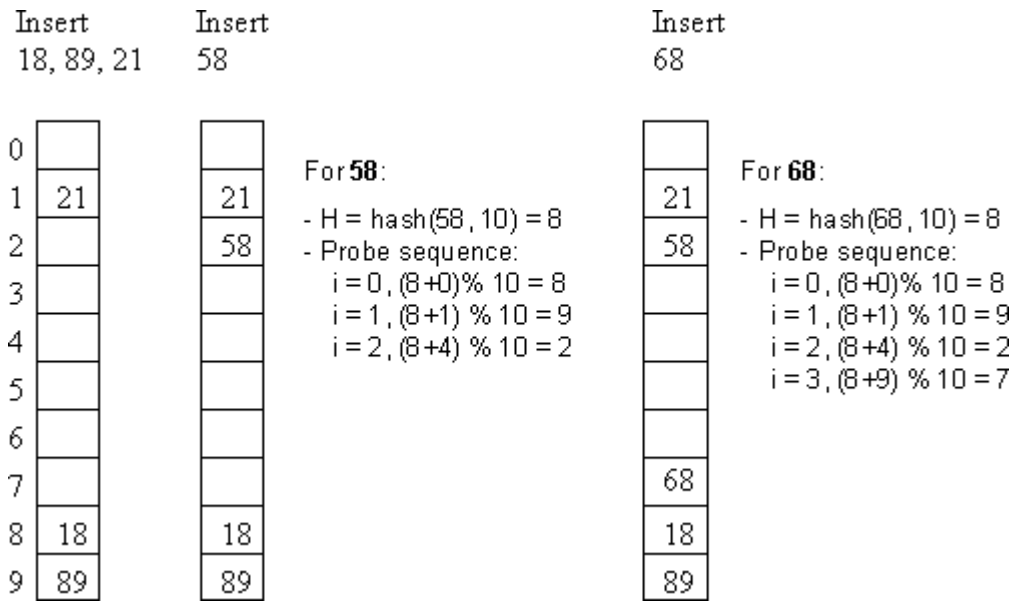
If slot hash(x) % S is full, then we try (hash(x) + 1*1) % S

If (hash(x) + 1*1) % S is also full, then we try (hash(x) + 2*2) % S

If (hash(x) + 2*2) % S is also full, then we try (hash(x) + 3*3) % S

.....

Sample Test Case:

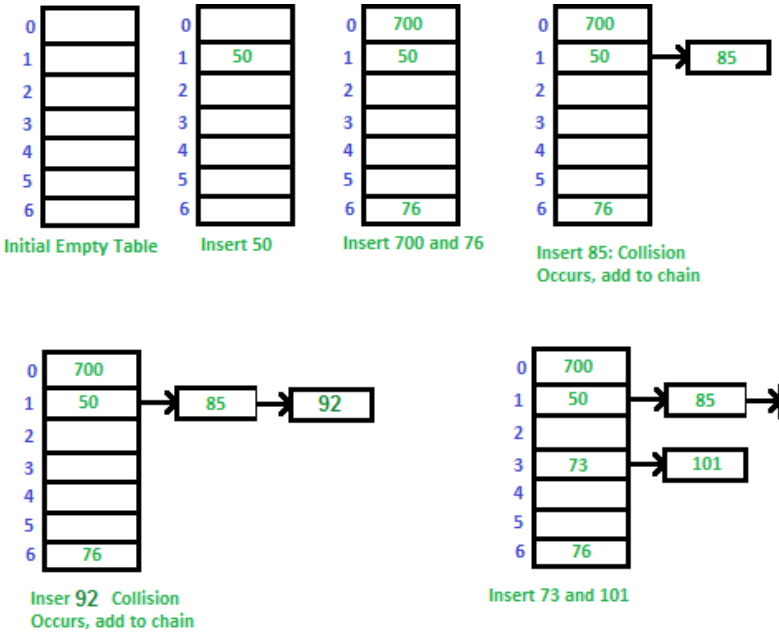


c) **Separate Chaining**: The idea is to make each cell of hash table points to a linked list of records that have same hash function value.

Let us store K keys into hash table of size S, where k<=n and k takes values from [1 to m], m>n.

Sample Test Case:

Let us consider a simple hash function as **–key mod 7**‖ and sequence of keys as 50, 700, 76, 85, 92, 73, 101.



5. Design, Develop and Implement a menu driven Program in C for the following. CO2

a) Operations on **STACK** of Integers (Array Implementation of Stack with maximum size **MAX**)

1. ***Push*** an Element on to Stack
2. ***Pop*** an Element from Stack
3. Demonstrate ***Overflow*** and ***Underflow*** situations on Stack
4. Display the status of Stack
5. Exit

b) Operations on **QUEUE** of Characters (Array Implementation of Queue with maximum size **MAX**)

1. Insert an Element on to QUEUE
2. Delete an Element from QUEUE
3. Demonstrate ***Overflow*** and ***Underflow*** situations on QUEUE
4. Display the status of QUEUE
5. Exit

Note: Support the program with appropriate functions for each of the above operations

6. Design, Develop and Implement a C program to do the following using a singly linked list. CO2

a) Stack- In single linked list store the information in the form of nodes .Create nodes using dynamic memory allocation method. All the single linked list operations perform based on Stack operations LIFO (last in first out).

A stack contains a top pointer. Which is ‐head‖ of the stack where pushing and popping items happens at the head of the list. first node have null in link field and second node link have first node address in link field and so on and last node address in ―top‖ pointer.

Stack Operations:

1. push() : Insert the element into linked list nothing but which is the top node of Stack.
2. pop() : Return top element from the Stack and move the top pointer to the second node of linked list or Stack.
3. peek(): Return the top element.
4. display(): Print all element of Stack.

b) Queue- All the single linked list operations perform based on queue operations FIFO (First in first out).

In a Queue data structure, we maintain two pointers, *front* and *rear*. The *front* points the first item of queue and *rear* points to last item.

1. **enQueue()** This operation adds a new node after *rear* and moves *rear* to the next node.
2. **deQueue()** This operation removes the front node and moves *front* to the next node.
3. Display() Display all elements of the queue.

Note: Sample node information: Student Data with the fields: USN, Name, Branch, Sem, PhNo.

7. Design, Develop and Implement a Program in C for the following CO2

a) Converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^(Power) and alphanumeric operands.

b) Evaluation of postfix expression with single digit operands and operators: +,-, *, /, %, ^

8) Design, Develop and Implement a menu driven Program in C for the following : CO2

a) Circular Queue

1. Insert an Element on to Circular QUEUE
2. Delete an Element from Circular QUEUE
3. Demonstrate *Overflow* and *Underflow* situations on Circular QUEUE
4. Display the status of Circular QUEUE
5. Exit

b) Priority Queue

1. Insert an Element on to Priority QUEUE
2. Delete an Element with highest priority from Priority QUEUE
3. Demonstrate *Overflow* and *Underflow* situations on Priority QUEUE
4. Display the status of Priority QUEUE
5. Exit

Support the program with appropriate functions for each of the above operations

9. Design, Develop and Implement a menu driven C program to Perform Operations on dequeue (double ended queue) using circular array. CO2

a) insertFront(): Adds an item at the front of Deque.
b) insertRear(): Adds an item at the rear of Deque.
c) deleteFront(): Deletes an item from front of Deque
d) deleteRear(): Deletes an item from rear of Deque
e) getFront(): Gets the front item from queue
f) getRear(): Gets the last item from queue
g) isEmpty(): Checks whether Deque is empty or not
h) isFull(): Checks whether Deque is full or not

Support the program with appropriate functions for each of the above operations

10. Design, Develop and Implement a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers CO3

a. Create a BST of N Integers: 13, 3, 4, 12, 14, 10, 5, 1, 8, 2, 7, 9, 11, 6, 18
b. Traverse the BST(either inorder, predorder or postorder)
c. Search the BST for a given element (KEY) and report the appropriate message
d. Exit

11. Design, Develop and Implement a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers CO3

a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2
b. Traverse the BST in Inorder, Preorder and Post Order using non-recursive functions
c. exit

12. Design, Develop and Implement a Program in C for the following operations on Graph(G) of Cities CO4

a. Create a Graph of N cities using Adjacency Matrix.
b. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method

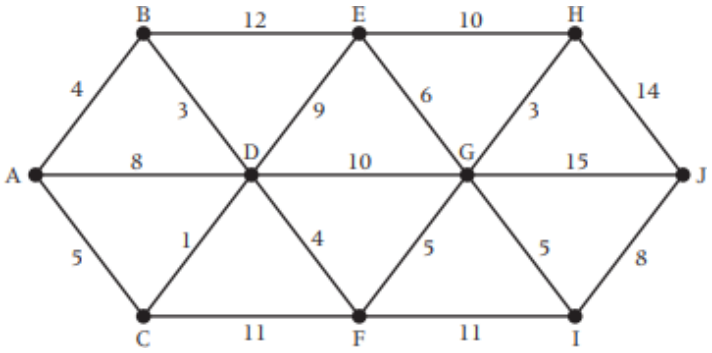
13. Design, Develop and Implement a C Program to the problem is to find shortest distances between every pair of vertices in a given edge weighted directed Graph using Warshall‟s Algorithm. The Graph is represented as Adjacency Matrix, and the Matrix denotes the weight of the edges (if it exists) else INF (1e7). CO4

Input:
The first line of input contains an integer **T** denoting the no of test cases. Then T test cases follow. The first line of each test case contains an integer V denoting the size of the adjacency matrix. The next V lines contain V space separated values of the matrix (graph). All input will be integer type.

Output:
For each test case output will be V*V space separated integers where the i-jth integer denote the shortest distance of ith vertex from jth vertex. For INT_MAX integers output INF.

Constraints:
1 <= T <= 20
1 <= V <= 100
1 <= graph[][] <= 500

14. Design, Develop and Implement a C Program to Find the shortest distance from A to J on the network below using Dijkstra‟s Algorithm CO4



Course Code 221	PROBABILITY , STATISTICS AND QUEUING THEORY [common to CSE& I.T.]	L 2	T 1	P 0	C 3

1. **Prerequisites** : Elementary knowledge of Set theory, Combinations , Calculus and basic Statistics
2. **Course Objective :**

The objective of this course is to provide the required mathematical support in real life problems and develop probabilistic models which can be used in several areas of science and engineering.

3. **Course Outcomes:** At the end of the course student should be able to:

CO - 1	Demonstrate basic principles of probability and understand a random variable that describe randomness or an uncertainty in certain realistic situation. It can be of either discrete or continuous type.
CO - 2	Comprehend concepts of discrete, continuous probability distributions and able to solve problems of probability using Binomial, Poisson, Uniform Distribution, Exponential Distribution, Normal distributions.
CO - 3	Compute simple correlation between the variables and fit straight line , parabola by the principle of least squares.
CO - 4	Analyze the statistical data and apply various small or large sample tests for testing the hypothesis.
CO - 5	Understand about different Queuing models and its applications.

Course Outcom	PO-a	PO-b	PO-c	PO-d	PO-e	PO-f	PO-g	PO-h	PO-i	PO-j	PO-k
CO - 1	**3**								**1**		**3**
CO - 2	**3**								**1**		**3**
CO - 3	**3**								**1**		**3**
CO - 4	**3**								**1**		**3**
CO - 5	**3**								**1**		**3**

Mapping of course outcomes with program outcomes:

DETAILED SYLLABUS:

UNIT- I : PROBABILITY & MATHEMATICAL EXPECTATIONS (12 Periods)

Introduction to Probability : Definition of Random Experiment, Events and Sample space, Definition of probability, Addition and Multiplication theorems, Conditional probability, Baye's Theorem, Simple Problems on Baye's theorem.

Introduction to Random variable : Discrete and Continuous random variables, Distribution function of random variable, Properties, Probability mass function, Probability density function, Mathematical expectation, Properties of Mathematical expectation, Moments, Moment generating function ,Mean and Variance.

Learning outcome: At the end of this unit, student will be able to Calculate probabilities using conditional probability, Rule of total probability and Bayes' theorem(L_3)

- Explain the concept of a random variable and the probability distributions(L_5)
- Express the features of discrete and continuous random variables and explain about probability mass ,density function and formulate the distribution functions. (L_5)
- Calculate the expected value of a random variable and moments and formulates the Moment Generating Function(L_3)

UNIT- II : PROBABILITY DISTRIBUTIONS (14 Periods)

Discrete Distributions: Binomial Distribution, Poisson distribution-Mean ,Variance, MGF and problems.

Continuous Probability Distributions: Uniform Distribution, Exponential Distribution, Memoryless property, Normal Distribution, properties of Normal Distribution, Importance of Normal Distribution, Area properties of Normal curve - MGF, Mean ,Variance and simple problems

Learning outcome: At the end of this unit, student will be able to

- Understand importance of discrete probability distributions Binomial, Poisson and solve the problems about these distributions (L_2)
- Understand importance of continuous distributions Exponential ,Uniform and Normal and Exponential Distribution and solve the problems about these distributions(L_2)
- calculate probabilities of events for these distributions using the probability function, probability density function or cumulative distribution function (L_3)

UNIT - III : CURVE FITTING , CORRELATION AND REGRESSION (10 Periods)

Curve Fitting : Principle of Least Squares, Method of Least Squares , Fitting of Straight lines, fitting of second degree curves and exponential curves

Correlation : Definition, Karl Pearson's Coefficient of Correlation Measures of correlation, Rank correlation coefficients.

Regression : Simple linear regression, Regression lines and properties.

Learning outcome: At the end of this unit, student will be able to

- Understand the concept of Principle of least squares for curve fitting of straight line ,second degree curve and exponential curve(L_2)
- Calculate Pearson‟s correlation coefficient, Spearman‟s rank correlation coefficient (L_3)
- Compute and interpret simple linear regression between two variables (L_3)
- Calculate regression coefficients and study the properties of regression lines (L_3)

UNIT- IV : TESTING OF HYPOTHESIS (14 Periods)

Introduction, Null hypothesis, Alternative hypothesis, Type –I, II errors, Level of significance, Critical Region. Confidence interval, one sided test, two sided test,

Small Sample Tests. : Students t - distribution , its properties; Test of significance difference between sample mean and population mean; difference between means of two small samples , F- Distribution, Test of equality of two population variances, Chi-square test of goodness of fit .

Large sample Tests : Test of Significance of Large Samples – Tests of significance difference between sample proportion and population proportion & difference between two sample proportions , Tests of significance difference between sample mean and population mean & difference between two sample means.

Learning outcome: At the end of this unit, student will be able to

- Formulate null & alternate hypothesis, identify type I & type II errors(L_6)
- Formulate, calculate and interpret hypotheses test for one parameter and to compare two parameters, for both large and Small samples, Z and T for one two samples (L_6)
- Perform and analyze hypotheses tests of means, proportions and variances using both one-and two-sample data sets (L_4)
- apply the appropriate Chi-Squared test for independence and goodness of fit(L_3)

UNIT- V : QUEUING THEORY (10 Periods)

Structure of a queuing system, Operating characteristics of queuing system, Transient and steady states, Terminology of Queuing systems, Arrival and service processes- Pure Birth-Death process Deterministic queuing models- M/M/1 Model of infinite queue, M/M/1 model of finite queue.

Learning outcome: At the end of this unit, student will be able to

- Explain pure birth and death process(L_5)
- Analyze M/M/1 model and solve traffic flow problems of M/M/1 model(L_4)
- understand various elements of a queuing system and each of its description(L_2)

TEXT BOOK :

1. Probability, Statistics and Random Processes by T.Veerarajan, Tata McGraw Hill Publications.

REFERENCE BOOKS:

1. Probability & Statistics with Reliability, Queuing and Computer Applications by Kishor S. Trivedi , Prentice Hall of India .
2. Higher Engineering Mathematics by Dr. B.S Grewal, Khanna Publishers
3. Probability and Statistics for Engineers and Scientists by Sheldon M.Ross, Academic Press
4. Fundamentals of Mathematical Statistics by S C Gupta and V.K.Kapoor

MICROPROCESSOR AND INTERFACING	
CSE 222	**Credits : 3**
Instruction **:** 3 Periods & 1 Tut/Week	Sessional Marks : 40
End Exam : 3 Hours	End Exam Marks : 60

Prerequisites:
Basic knowledge of Digital Logic Design and Computer Organization.

Course Objectives:

1. The objective of this course is to become familiar with the architecture and the instruction set of an Intel microprocessors.
2. Assembly language programming will be studied for practical implementation of the programs in trainer kit.
3. Learning the concept of interfacing various I/O peripherals like Keyboard/Display, stepper motor etc., with microprocessors using 8255 PPI.
4. To provide solid foundation on interfacing the external devices to the processor according to the user requirements to create novel products and solutions for the real time problems.
5. The accompanying lab is designed to provide practical hands-on experience with microprocessor software applications and interfacing techniques

Course Outcomes

By the end of the course, the student will be able to:	
1.	Describe the architecture and pin configuration of 8085 Microprocessors and the significance of Addressing modes, timing diagrams and analyze the working of instruction set.
2.	Demonstrate the programming knowledge for practical implementation of assemble level programming using instruction set of 8085.
3.	Analyze the working of 8085 interfacing with co-processors are 8255, 8251, 8253, 8259, 8279 and External I/O devices.
4.	Describe the architecture and pin configuration of 8086 Microprocessors and the significance of Addressing modes, Segmented memory and Min-Max mode operations
5.	Demonstrate the programming knowledge for practical implementation of assemble level programming using instruction set of 8086.

Mapping of Course Outcomes with Program Outcomes:

Mapping		PO												PSO	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO	**1**	3	1							3	1			1	2
	2	3	3	2	2	2				1	1				1
	3	3	3	3						2					
	4	2	1	2										1	
	5	3	3	2	2	2				1	1				1

SYLLABUS

UNIT I **15 periods**

Introduction to 8085 Microprocessor

Introduction to Microprocessors and Microcomputers, Internal Architecture and Functional / Signal Description of typical 8-bit µP. 8085, Instruction Set, types of Instructions, Addressing modes of 8085 and Timing Diagrams of 8085 µP.

Learning Outcome: At the end of this Unit the students will be able to

- Draw and describe the basic architecture of 8085 and the functional description of 8085.
- List the Instruction set; state the addressing modes and timing diagrams of 8085.

UNIT II **10 periods**

8085 µP Assembly Language Programming

Introduction to Assembly Language Programming Techniques: Looping, Counting, and Indexing, Counter and timing Delays, Stack and Subroutines, Code Conversions, BCD Arithmetic operations , 16-bit data Operations, Interrupts and Interrupt Service Routines.

Learning Outcome: At the end of this Unit the students will be able to

- Develop the assembly language programs using various programming techniques
- Analyze the simple programs of call instructions, sorting, and string manipulations.

UNIT-III **15 periods**

Interfacing Peripheral ICs to Intel 8085

Programmable peripheral interface (8255A), Programmable communication interface (8251), Programmable Interval timer (8253 and 8254), Programmable Interrupt controller (8259), Programmable Keyboard / Display controller (l 8279).

Learning Outcome: At the end of this Unit the students will be able to

- Illustrate how the different peripherals (8255, 8251, 8253, 8259, 8279.) are interfaced with Microprocessor.
- Describe the control word formats of all Programmable peripheral interfaces.

UNIT IV **12 periods**

Introduction to 8086 Microprocessor

Internal Architecture and Functional/Signal Description of 8086/8088, Segmented Memory, Maximum-Mode and Minimum-Mode Operation and Addressing Modes of 8086.

Learning Outcome: At the end of this Unit the students will be able to

- Describe the modes and functional block diagram of 8086 along with pins and their functions.
- Develop the assembly language programs using various programming techniques.

UNIT V

8086 µP Assembly Language Programming

8 periods

Instruction Set and Timing Diagrams, Interrupts and Interrupt Service Routines, Assembler Directives, Loops Procedures, Modular programming and Macros, .COM and .EXE formats.

Learning Outcome: At the end of this Unit the students will be able to

- Used the programming techniques of 8086 to build programs using instruction set.
- List, describe and use different types of instruction, directives and interrupts.

Case Study: Introduction to Pentium and Multi-Core Processors and Arduino processors.

TEXT BOOKS:

1. Ramesh S. Gaonkar, ―Microprocessor Architecture, Programming, and Applications with the 8085‖ Penram International, 6th Edition.

2. John E.Uffenbeck, ―The 80x86 Family, Design, Programming and Interfacing 3rdEdition, Pearson Education Inc.‖, 2002.

REFERENCE BOOKS:

1. BARRY B. BREY, –The Intel Microprocessors 8086/8088, 80186/80188,80286,80386 and 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, Pentium 4, Architecture, Programming and Interfacing‖, Pearson Education Inc., 2003,6thEdition.
2. Walter A. Tribel and Avtar Singh,‖ The 8088 and 8086 Microprocessors, Programming, interfacing, Software, Hardware, and Applications‖, Pearson Education Inc., 2003, 4thEdition.
3. Douglass V. Hall, –Microprocessors and Interfacing, Programming and Hardware| , TMH Edition, 1999, 2ndEdition
4. Sanjay K Bose, –Hardware and Software of Personal Computers‖, New Age International (P) Ltd., 1991.
5. A.P. Mathur, –Introduction to Microprocessor|, Tata McGraw-Hill Education, 1989.
6. YU-Cheng Liu & Glenn A Gibson, –Microprocessor System, Architecture Programming & Design‖.

ONLINE WEB RESOURCES:

1. https://www.tutorialspoint.com/microprocessor/index.htm
2. https://swayam.gov.in/nd1_noc20_ee11/preview
3. https://medium.com/@harshityadav95/microprocessor-and-interfacing-ef9c2b116382
4. http://pages.hmc.edu/harris/cmosvlsi/4e/lect/lect24.pdf
5. https://www.arduino.cc/

Prepared By: G Gowripushpa, P. N Srinivas

OPERATING SYSTEMS	
CSE 223	**Credits : 3**
Instruction **:** 3 Periods & 1 Tut/Week	Sessional Marks : 40
End Exam : 3 Hours	End Exam Marks : 60

Prerequisites: Basic programming language and Computer Organization.

Course Objectives:

1. To understand the main components of operating system and their functions.
2. To understand the basic concept of shell programming.
3. To learn the mechanism of an operating system as process manager, memory manager, device manager and file manager.
4. To understand the concept of protection related to operating system.

Course Outcomes:

1. Illustrate the structure of OS, Functionality and services provided by the OS. Analyse the concept of shell programming, process state and state transitions.
2. Implement the CPU Scheduling algorithms (Pre-emptive and Non Pre-emptive). Demonstrate the concept of Process synchronization.
3. Demonstrate the concept of resource allocation. Apply and analyze the various memory management mechanisms for contiguous and non contiguous memory.
4. Demonstrate the structure and organization of file systems and analyze the implementation of file systems.
5. Analyse the secondary storage structure, protection of the system.

Mapping of Course Outcomes with Program Outcomes:

Mapping		**PO**												**PSO**	
		1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO	1	1	1	1		2				1	1		2		
	2	2	2	2	2	2	2	2		2	1	2	2	1	1
	3	2	2	2	2	2	2	2		2	1	2	2	1	1
	4	2	2	2	1	2	1	1		2	1	1	2	1	1
	5	2	2	2	1	2	1	1		2	1	1	2	1	1

COURSE CONTENTS

UNIT I **(12 Periods)**

Introduction to OS: Operating system Definition, Operating system Functionalities, Types of Operating system, operating system structures, system calls, system programs.-

Introduction to Shell Programming: Commands and Shell script.

Processes: Process concept, Process scheduling, Operations on processes, Inter process communication, Communication in client-server systems.

Threads: Overview, Multithreading models.

Learning outcomes: At the end of this Unit, Students are able to

1. Define the responsibilities of an operating system and implement the basic shell programs.
2. Demonstrate the different modes of communication among processes and multi threading models.

UNIT II **(12 Periods)**

CPU Scheduling: Scheduling criteria, Scheduling algorithms, Algorithm Evaluation.
Process Synchronization: The critical-section problem, Peterson‟s solution, Synchronization hardware, Mutex Locks, Semaphores, Classic problems of synchronization, Monitors.
Case Study: Linux operating system: Process Management.

Learning outcomes: At the end of this Unit, Students are able to

1. Analyze the CPU scheduling algorithms and their performance evaluation.
2. Implement the different solutions for process synchronization.

UNIT III **(12 Periods)**

Deadlock: System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock.
Memory Management: Background, Swapping, Contiguous memory allocation, Segmentation, Paging, Structure of the page table.
Virtual Memory: Background, Demand paging, Page replacement, Allocation of frames, Thrashing.
Case Study: Linux operating system: Memory Management.

Learning outcomes: At the end of this Unit, Students are able to

1. Define the concept of deadlock and Identify the different ways to handle deadlock like prevention, detection, avoiding and recovery.
2. Distinguish between contiguous and non-contiguous memory allocation methods in memory management.

UNIT IV **(12 Periods)**

File Systems Interface: File concept, Access methods, Directory structure, File system mounting, File Sharing, Protection.
Implementing File-Systems: File system structure, File system implementation, Directory implementation, Allocation methods, Free-space management, Efficiency and performance,
Recovery.

Learning outcomes: At the end of this unit, students are able to

1. Demonstrate the concept of file system, various file access methods and Protection in files.
2. Identify and implement the file system and recovery.

UNIT V **(12 Periods)**

Secondary Storage Structure: Mass storage structures, Disk structure, Disk attachment, Disk scheduling, Disk management, Swap space management.
Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights.

Learning outcomes: At the end of this unit, students are able to

1. Demonstrate the concept of mass storage structures and Analyze the various disk scheduling algorithms
2. State the goal and principles of protection and implement the access matrix.

TEXT BOOKS

1. Silberschatz, Galvin and Gagne, –Operating System Principles‖, 9th Edition, Wiley India Pvt Ltd, 2015.
2. Sumitabha Das, ―Unix Concepts and Applications‖, 4th Edition. TMH, 2006.
3. Yashwanth Kanitkar, –Unix Shell programming‖, 1st Edition, BPB Publisher, 2010.

REFERENCES

1. Andrew S. Tanenbaum, –Modern Operating Systems‖, 4th Edition, Pearson Education, 2015.
2. William Stalling, –Operating Systems: Internals and Design Principles‖, 9th edition, PHI, 2018.
3. Harvey M. Deitel, ―Operating Systems‖, 3rd Edition, Pearson Education, 2004.
4. M.G.Venkateshmurthy, –Introduction to Unix and Shell Programming‖, 5th Edition, Pearson Education India, 2009.
5. N.B Venkateswarlu, –Advanced Unix programming‖, 2nd Edition, BS Publications, 2010.

WEB REFERENCES:

1. https://opensource.com/resources/linux
2. https://nptel.ac.in/courses/106/106/106106144/
3. http://openbookproject.net/courses/intro2ict/system/os_intro.html
4. https://en.wikipedia.org/wiki/Xv6.
5. https://nptel.ac.in/content/storage2/courses/106108101/pdf/PPTs/Mod_13.pdf

---

Prepared By: Dr.K.S.Deepthi, Mr.K.Chandra Sekhar

COMPUTER NETWORKS	
CSE 224	**Credits : 3**
Instruction **:** 3 Periods & 1 Tut/Week	Sessional Marks : 40
End Exam : 3 Hours	End Exam Marks : 60

Prerequisites:

Good Knowledge on Data Communications and Networking technologies

Good Knowledge of Programming Languages (such as C, JAVA) for better understanding of network programming.

Course Objectives:

The course should enable the students:

Course Objectives:

- To make the student understand the contemporary technologies in network protocols and network architecture.
- To acquire the knowledge on design principles of network infrastructure
- To learn and understand the design issues in framing and error handling
- To gain a sufficient knowledge on addressing the nodes in the network and connecting them using the network level protocols.
- To make them familiarize with different application layer protocols and network management elements.

Course Outcomes:

the end of the course, the student will be able to:	
1.	Identify the Network Components required to build different types of network and Hardware devices, signal modulation techniques at Physical Layer.
2.	Apply appropriate error control, flow control techniques and MAC protocols.
3.	Contrast connection-oriented and connectionless services for datagram. Identify Routing Algorithms and congestion control techniques.
4.	Trace the flow of information from one node to another in the network.
5.	Classify the network services and analyze how to manage the network.

Mapping of Course Outcomes with Program Outcomes:

Mapping		**PO**												**PSO**	
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**1**	**2**
CO	**1**	**3**	**3**	**2**	**1**	**3**				**2**	**1**		**3**	**1**	**2**
	2	**3**	**3**	**2**	**2**	**1**				**2**	**2**		**2**	**1**	**2**
	3	**3**	**3**	**2**	**2**	**1**				**2**	**2**		**2**	**1**	**2**
	4	**3**	**3**	**2**	**2**	**1**				**2**	**1**		**2**	**1**	**2**
	5	**3**	**3**	**2**	**2**	**1**		**2**		**2**	**2**		**2**	**1**	**2**

SYLLABUS

UNIT-I: **10 periods**

Introduction to Computer Networks:

Introduction to Data Communication, Network- Components, Categories and classification, Network Topologies, Reference Models-ISO/OSI, TCP/IP, Transmission Media, ARPANET, Network related tools and commands.

Physical Layer:

Hardware devices, Signal Representation, Modulation and Demodulation Techniques, Spread Spectrum,

Security issues, Switching Techniques: Circuit Switching, Packet Switching, Message Switching.
Learning Outcome: At the end of this Unit the student will be able to

- Identify the various design parameters and their influence on link utilization performance
- Identify the Network Components required to build different types of network and Hardware devices, signal modulation techniques at Physical Layer.

UNIT-II: **10 periods**

Data Link Layer:

LLC: DataLink Layer Design Issues, Error-Error Detection,Control and Correction- Parity,LRC,CRC-Hamming Code, Flow Control- Sliding Window Protocols-Go Back N, Selective Repeat, HDLC.

Medium Access Control Layer:

Channel Allocation Problems- Static and Dynamic, MAC protocols: ALOHA, CSMA- CSMA/CD, CSMA/CA, IEEE 802.3- Ethernet Frame Format, Backoff Algorithm, IEEE 802.4, IEEE 802.11.
Learning Outcome: At the end of this Unit the student will be able to

- Compare Scenario-Specific Protocols for Wired Communication.
- Apply appropriate error control, flow control techniques and MAC protocols.

UNIT-III: **14 periods**

Network Layer:

Packet Switching and Datagram Approach, Protocols-ARP,DHCP,IPv4, IPv6, ICMP, Fragmentation, Routing Algorithms(including Border Gateway). Subnetting Concept, Network Address Translation, Congestion Control Techniques, Tunneling, Security issues.
Learning Outcome: At the end of this Unit the student will be able to

- Apply Different technologies to Connection oriented and connectionless for real time.
- Contrast connection-oriented and connectionless services for datagram

UNIT-IV: **14 periods**

Transport Layer:

Transport Layer Responsibilities, Reliable end to end Protocols: (TCP, UDP), TCP and UDP Protocol‟s Congestion and flow control mechanisms, TCP Extensions, QOS, Security issues.
Learning Outcome: At the end of this Unit the student will be able to

- Evaluate the challenges in end to end Data delivery
- Trace the flow of information from one node to another in the network.

UNIT-V: **12 periods**

Application Layer:

Protocols: SMTP, FTP, TFTP, BOOTP, HTTP/HTTPS, MIME, POP.

Network Services and Management:

Client-Server and Peer-to-Peer Architectures-www-e-mail-DNS-VoIP, SNMP management.

Basics of Network Security: Attacks, Symmetric and Asymmetric Cryptography.
Learning Outcome: At the end of this Unit the student will be able to

- Design efficient network services using network management strategies
- Classify the network services and analyze how to manage the network.

Text Books:

1. Andrew S. Tanenbaum, –*Computer Networks*|, 5th edition, Prentice-Hall Publisher
2. Behrouz A Forouzan –*Data Communications and Networking*| , 4th Edtion, Tata McGraw-Hill.

Reference Books:

1. William Stallings, –*Data and Computer Communications*|. 7th edition, Pearson Education
2. J F Kurose, K W Ross, –*Computer Networking: A Top-Down Approach"*, 5th Edition, Addison-Wesley.

Web Resources:

1. https://nptel.ac.in/courses/106/105/106105081/
2. https://swayam.gov.in/nd1_noc20_cs23/preview
3. https://www.gatevidyalay.com/computer-networks/

Prepared By: Mr. G.V. Eswara Rao, Mrs.S.S.N.L.Priyanka

COMPUTER GRAPHICS	
CSE 225	**Credits : 3**
Instruction **:** 3 Periods & 1 Tut/Week	Sessional Marks : 40
End Exam : 3 Hours	End Exam Marks : 60

PREREQUISITES:

Elementary knowledge in C programming, Solving mathematical expressions, algorithm or pseudo code.

COURSE OBJECTIVES:

- Understand the applications in the real world and the graphics systems used in developing graphics.
- Exploration of fundamental concepts in 2D and 3D computer graphics.
- Learn two dimensional and three dimensional computer graphics with comprehend advanced software tools of computer graphics

COURSE OUTCOMES:

the end of the course, the student will be able to:	
1.	Explain computer graphics, applications and contemporary terminology, hardware
2.	Design 2D objects using various algorithms.
3.	Apply geometric and viewing transformations on 2D objects.
4.	Design 3D objects and apply geometric and viewing transformations on 3D objects.
5.	Compare visible surface methods.

Mapping of Course Outcomes with Program Outcomes:

Mapping		PO												PSO	
		1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO	1	1	1										1		
	2	3	2	2	2					2	1		1		2
	3	3	2	2	2					2	1		1		2
	4	3	2	2	2					2	1		1		2
	5	2	1	1						1	1				2

SYLLABUS

UNIT-I **9periods**

Introduction: Basics of computer graphics, Applications

Over view of Graphics systems: Video Display Devices, Raster Scan systems, Random scan systems, Graphics monitors and workstations, Input devices, Graphics software.

Learning Outcomes : At the end of this unit, student will be able to

1. Express about the application in the real world and the computer Graphics.
2. Summarize the different graphic systems

UNIT-II **15 periods**

Output primitives & its attributes: Points and Lines-Line Drawing Algorithms, Loading the Frame buffer, Line function, Circle Generating Algorithms, Ellipse Generating Algorithms, Filled Area Primitives, Filled Area Functions, Cell Array, Character Generation.

Attributes of Output Primitives: Line and Curve Attributes, Color and Gray scale levels, Area Fill Attributes, Character Attributes, Bundled Attributes, Anti-aliasing.

Learning Outcomes : At the end of this unit, student will be able to

1. Observe various 2d output primitive and algorithm
2. Interpret the attributes of output primitives

UNIT –III **15 periods**

Two Dimensional Geometric Transformations: Basic Transformations, Matrix Representations, Homogeneous Coordinates, Composite Transformations, Other Transformations, Transformations between Coordinate Systems.

Two Dimensional Viewing: The Viewing Pipeline, Viewing Coordinate Reference Frame, Window to Viewport Coordinate Transformation, Two Dimensional Viewing Functions, Clipping Operations, Point Clipping, Line Clipping, Polygon Clipping, Curve Clipping, Text and Exterior Clipping.

Learning Outcomes : At the end of this unit, student will be able to

1. Apply 2-dimensional geometric transformation to the real world scenario
2. Evaluate various clipping algorithms and outline 2D viewing transformation

UNIT-IV **12 periods**

Three Dimensional Concepts and Object representations & Transformation : 3D display methods, 3D Graphics, Polygon Surfaces, Curved Lines and Surfaces, Quadratic Surfaces, Super Quadrics, Blobby Objects, Spline Representations, Cubic Spline methods, Bezier Curves and Surfaces, B Spline Curves and Surfaces.

Three Dimensional Geometric and Modeling Transformations: Translation, Rotation, Scaling, Other Transformations, Composite Transformations.

Three Dimensional Viewing: Viewing Pipeline, Viewing Coordinates, Projections, View Volumes, General Projection Transformations, Clipping

Learning Outcomes : At the end of this unit, student will be able to

1. Review the 3d object representations and the analyze various surface representation.
2. Compare the 2D and 3D geometric and modeling transformations and explain the 3D viewing transformation.

UNIT-V **9 periods**

Visible Surface Detection & Hidden Surface Detection Methods: Classification of visible, Surface detection algorithms, Back face method, Depth buffer method, Scan line method, Depth Sorting Method, Z-buffer method, Area sub-division method, Comparison of hidden surface methods.

Polygon Rendering Methods: Constant-Intensity Method, Gouraud Method, Phong Method

Learning Outcomes : At the end of this unit, student will be able to

1. Compare and differentiate visible surface detection methods
2. Differentiate various polygon rendering methods.

Text Books:

1.Computer Graphics C Version by Donald Hearn & M. Pauline Baker Pearson Education, New Delhi, 2004

References Books:

1. Procedural Elements for Computer Graphics by David F. Rogers, Tata McGraw Hill Book Company, New Delhi, 2003

2. Computer Graphics: Principles & Practice in C by J. D. Foley, S. K Feiner, A Van Dam F. H John, Pearson Education, 2004

Web References:

1. http://nptel.ac.in/courses/106106090/
2. https://www.coursera.org/courses?languages=en&query=computer+graphics
3. https://courses.edx.org/courses/BerkeleyX/CS-184.1x/2013_October/syllabus/

,,

Compiled By : Mrs G V Gayathri, Asst Professor,CSE Dept
Mrs G Santoshi, Asst Professor,CSE Dept

FORMAL LANGUAGES AND AUTOMATA THEORY	
CSE 226	**Credits: 3**
Instruction**:3** Periods & 1 Tut/Week	Sessional Marks : 40
End Exam : 3 Hours	End Exam Marks : 60

Prerequisites:

- The students are expected to have a strong background in the fundamentals of discrete mathematics (symbolic logic, set, induction, number theory, summation, series, combinatory, graph, recursion, basic proof techniques, etc.), algorithms and data structures.
- Some knowledge of programming languages, programming, and computer architecture will be helpful.

Course Objectives:

The course should enable the students:

- To provide introduction to some of the central ideas of theoretical computer science from the perspective of formal languages.
- To introduce the fundamental concepts of formal languages, grammars and automata theory.
- Classify machines by their power to recognize languages.
- Employ finite state machines to solve problems in computing.
- To understand deterministic and non-deterministic machines.
- To understand the differences between decidability and undecidability.

Course Outcomes:

By the end of the course, the student will be able to:	
1.	Acquire a fundamental understanding of the core concepts in automata theory, construct DFAand NFA. Ability to transform between equivalent finite automata, Construct Epsilon-NFA and transform between equivalent finite automata
2.	Understand the power and the limitations of regular expressions and design regular expressions. Compute transformation between finite automata and regular expressions
3.	Describe and construct Context Free Grammar and Pushdown Automata, transformation between them
4.	Construct and analyze the use and properties of Turing machines performing simple tasks, with recent trends and applications in the area of finite state machines
5.	Understand the concepts of recursively enumerable languages and undecidability problem.

Mapping of Course Outcomes with Program Outcomes:

S.NO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO 1	3	3	2	3	1	1	0	1	1	3	0	2	3	1
CO 2	2	2	3	2	0	0	0	1	1	2	0	2	2	1
CO 3	2	2	3	2	0	0	0	1	1	2	0	2	3	1
CO 4	2	3	3	2	0	0	0	1	1	2	0	3	2	1
CO5	2	3	3	3	0	0	0	1	1	2	0	3	3	2

SYLLABUS

UNIT-I: **12 periods**

FINITE AUTOMATA (FA): Introduction, Deterministic Finite Automata (DFA) -Formal definition, simpler notations (state transition diagram, transition table), language of a DFA.

NONDETERMINISTIC FINITE AUTOMATA (NFA)- Definition of NFA, language of an NFA, Equivalence of Deterministic and Nondeterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of Deterministic Finite Automata, Finite automata with output (Moore and Mealy machines) and Inter conversion.

LEARNING OUTCOMES:

1. An ability to design grammars and automata (recognizers) for different language classes.
2. An ability to prove and disprove theorems establishing key properties of formal languages and automata.

UNIT-II: **12 periods**

REGULAR EXPRESSIONS (RE): Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA are to Regular Expressions, Converting Regular Expressions to Automata, and applications of Regular Expressions.

REGULAR GRAMMARS: Definition, regular grammars and FA, FA for regular grammar, Regular grammar for FA. Proving languages to be non-regular -Pumping lemma, applications, and Closure properties of regular languages.

LEARNING OUTCOMES:

1. Design Finite Automata's for different Regular Expressions and Languages.
2. Understand the Pumping lemma for proving that languages are not regular.

UNIT-III: **12 periods**

CONTEXT FREE GRAMMER (CFG): Derivation Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings. Ambiguity in CFG"s, Minimization of CFG"s, CNF, GNF, Pumping Lemma for CFL"s, Enumeration of Properties of CFL (Proof's omitted).

LEARNING OUTCOMES:

1. To construct context free grammar for various languages.
2. Describe the language accepted by automata or generated by a regular expression or a context-free grammar.

UNIT-IV: **12 periods**

PUSHDOWN AUTOMATA: Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack and its Equivalence, Equivalence of CFG and PDA.

TURING MACHINES (TM): Formal definition and behavior, Languages of a TM, TM as accepters and as a computer of integer functions, Types of TMs.

LEARNING OUTCOMES:

1. To solve various problems of applying normal form techniques, push down automata and Turing Machines.
2. Construct a pushdown automaton for a given context-free language
3. Construct a total Turing machine deciding a given problem

UNIT-V: **12 periods**

RECURSIVE AND RECURSIVELY ENUMERABLE LANGUAGES (REL): Properties of recursive and recursively enumerable languages, Universal Turing machine, The Halting problem, Undecidable problems about TMs. Context sensitive language and linear bounded automata (LBA), Chomsky hierarchy, Decidability, Post's correspondence problem (PCP), undecidability of PCP.

LEARNING OUTCOMES:

1. Understand the basic results on computability, including undecidable problems such as the halting and Post correspondence problems, and their significance.
2. Define the various categories of languages and grammars in the Chomsky hierarchy.

TEXT BOOKS

1. –Introduction to Automata Theory Languages and Computation‖. Hopcroft H.E. and Ullman J. D. Pearson Education.

REFERENCE BOOKS

1. Theory of Computer Science – Automata languages and computation -Mishra and Chandrashekaran, 2nd edition, PHI.
2. Introduction to languages and the Theory of Computation, John C Martin, TMH, 4th edition.

ONLINE WEB RESOURCES

1. https://nptel.ac.in/courses/111103016/

Prepared By: Dr.K. Suresh, Mr.G.Sathar

MICROPROCESSOR AND INTERFACING LAB	
CSE 227	**Credits : 1.5**
Instruction **:** 3 Periods / Week	Sessional Marks : 50
End Exam : 3 Hours	End Exam Marks :50

Prerequisites:

Basic knowledge of Digital Logic Design and Computer Organization.

Course Objectives:

1. Developing of assembly level programs and providing the basics of the processors
2. To provide solid foundation on interfacing the external devices to the processor according to the user requirements to create novel products and solutions for the real time problems
3. To assist the students with sufficient knowledge on the interrupts and working with interrupt driven I/O for communication with external devices.

Course Outcomes:

By the end of the course, the student will be able to:	
1.	Possessed a better command over the instruction of set of 8085 and 8086 microprocessor for programmatically deployment.
2.	Demonstrate the interfacing of 8085 microprocessor with external I/O devices through 8255 PPI.
3.	Analyze the internal communication of microprocessor with the external devices through the interrupts and working with various types of vector interrupts
4.	Students will possess the knowledge to design and develop a working prototype with various simulators and emulators that they have used throughout the lab sessions.

Mapping of Course Outcomes with Program Outcomes:

Mapping		PO												PSO	
		1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO	1	3	1	3	1	1			2	2	1			2	2
	2	2	1	2		1				1	1				1
	3	1	1	2						1					1
	4	2	1	2		1				2					2

ASSEMBLY LANGUAGE PROGRAMMING:

1. **8085 ASSEMBLY LANGUAGE PROGRAMMING** According to theory course using the following: **Using Keyboard Monitor of 8085µP Trainer.** 3 weeks
2. **INTERFACING WITH 8085 TRAINER** 4 weeks
   - 2.1 8255 STUDY CARD SCENARIOS (I/O and BSR MODE OPERATIONS)
   - 2.2 8255 MODES using HARDWARE INTERRUPTS
   - 2.3 KEYBOARD/DISPLAY INTERFACE
3. **INTERFACING WITH PC** 4 weeks
   - 3.1 TRAFFIC LIGHT CONTROLLER
   - 3.2 STEPPER MOTOR CONTROLLER
   - 3.3 LOGIC CONTROLLER

4. **8086 ASSEMBLY LANGUAGE PROGRAMMING** According to theory course using the following: **PC Assembler using TASM or MASM, TD or SYMDEB or CVD (Code View debugger)** 2 weeks

TEXT BOOKS:

1. Ramesh S. Gaonkar, ―Microprocessor Architecture, Programming, and Applications with the 8085‖ Penram International, 6th Edition.
2. John E.Uffenbeck, ―The 80x86 Family, Design, Programming and Interfacing 3rdEdition, Pearson Education Inc.‖, 2002.

REFERENCE BOOKS:

7. BARRY B. BREY, ―The Intel Microprocessors 8086/8088, 80186/80188,80286,80386 and 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, Pentium 4, Architecture, Programming and Interfacing‖, Pearson Education Inc., 2003,6thEdition.
8. Walter A. Tribel and Avtar Singh,‖ The 8088 and 8086 Microprocessors, Programming, interfacing, Software, Hardware, and Applications‖, Pearson Education Inc., 2003, 4thEdition.
9. Douglass V. Hall, ―Microprocessors and Interfacing, Programming and Hardware‖ , TMH Edition, 1999, 2ndEdition

ONLINE WEB RESOURCES:

6. https://swayam.gov.in/nd1_noc20_ee11/preview
7. http://www.brcmcet.edu.in/downloads/files/n552f41719937b.pdf
8. http://anandsrys.weebly.com/uploads/2/3/9/6/23968450/mpi_lab_iii_year_cse_180.pdf

---

Prepared By: G Gowripushpa, P. N Srinivas

OPERATING SYSTEM LAB	
CSE 228	**Credits : 1.5**
Instruction **:** 3 Periods per week	Sessional Marks : 50
End Exam : 3 Hours	End Exam Marks : 50

Prerequisites: Basic programming language

Course objective:

1. To learn and execute the basic shell script, UNIX commands and system calls.
2. To understand and implement the process, memory and file management.
3. To solve the problems related to process synchronization.

Course Outcomes:

CO1: Execute the Unix Shell programming on the given system configuration.

CO2: Learn the various services provided by the system calls.

CO3: Simulate the process scheduling, process synchronization, deadlock avoidance and detection algorithms.

CO4: Simulate memory management techniques and file handling.

Mapping of Course Outcomes with Program Outcomes:

Mapping		**PO**												**PSO**	
		1	2	3	4	5	6	7	8	9	10	11	12	1	2
CO	1	1	0	0	1	2	1	1	0	2	1		2	2	1
	2	1	0	0	1	2	1	1	0	2	1		2	2	1
	3	2	2	2	2	2	2	2	0	2	1	2	2	2	2
	4	2	2	2	1	2	2	2	0	2	1	1	2	2	2

LIST OF SAMPLE PROGRAMS

Write a Program for the following

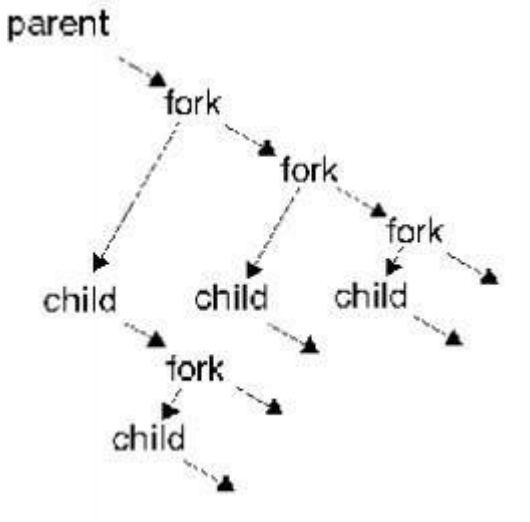
1. Implement basic shell commands. (CO1)

2. Shell programming: Simple logic programs. (CO1)

   i) Write a menu driven script using the select statement to print calories for food items such as pizza, burger, Salad, Pasta etc.

   ii) Write a shell script that, given a file name as the argument will count vowels, blank spaces, characters, number of line and symbols.

3. i) Analyze the below situation and develop a program for creating processes as required. Print the PID's of each process in a convenient way to understand. (CO2)



ii) Write a program to create two processes P1 and P2. P1 takes a string and passes it to P2. P2 concatenates the received string with another string without using string function and sends it back to P1 for printing. (CO2)

4. CPU Scheduling Algorithms (CO3)

i) A washing machine which require the process to be executed sequentially. Consider the processes P1, P2, P3, P4 whose arrival times are 1, 5, 9, 10 and burst times are 4, 3, 5, 2 respectively. Implement an appropriate algorithm. Find the CPU idle time, so that the water can be supplied during that period of time.

ii) Implement shortest job first for the following data:

Consider the following set of processes, CPU burst time, Arrival time. Calculate the average waiting time, average response time and average turnaround time.

Process	Burst Time	Arrival Time
P1	3	0
P2	6	2
P3	4	4
P4	5	6
P5	2	8

iii) Implement Round Robin for the following data

Consider the following set of processes and length of the CPU burst time given in milliseconds.

Process	Burst Time
P1	10
P2	1
P3	2
P4	1
P5	5

The processes are assumed to have arrived in the order P1, P2, P3, P4, P5 all at time 0 and time quantum in RR=1.Calculate the average waiting time, response time and turnaround time.

5. Develop a program to provide synchronization among the 5 philosophers in Dining Philosophers problem using semaphore. (CO3)

6. Develop a program to provide synchronization among the producer and consumer processes in producer – consumer problem using a monitor. (CO3)

7. Consider the following data: (CO3)

Process	Allocation A B C D	Max A B C D	Available A B C D
P1	0 0 1 2	0 0 1 2	2 1 0 0
P2	2 0 0 0	2 7 5 0	
P3	0 0 3 4	6 6 5 6	
P4	2 3 5 4	4 3 5 6	
P5	0 3 3 2	0 6 5 2	

i) Calculate the need matrix
ii) Is this system currently in a safe or unsafe state?
iii) Is the system currently deadlock or not.
iv) Which process, if any, or may become deadlocked?

8. Consider the following scenario: A process has been allocated 3 page frames. Assume that none of the pages of the process are available in the memory initially. The process makes the following sequence of page references (reference string): 1, 2, 1, 3, 7, 4, 5, 6, 3, 1, 2, 4, 6, 3, 1. Find out a page replacement policy which gives least number of page faults. (CO4)

9. Simulate the Virtual Memory concept. (CO4)

10. Implement the first fit and best fit algorithm in memory management. (CO4)

11. Simulate the Contiguous file allocation method. (CO4)

12. Implement bit map for the following scenario. (CO4)
For a memory of size 32 blocks ,the allocated blocks are 2,3,4,5,8,9,10,11,12 and display the bitmap pattern.

REFERENCES:

1. Sumitabha Das, –Unix Concepts and Applications‖, 4th Edition. TMH, 2006.
2. Yashwanth Kanitkar, –Unix Shell programming‖, 1st Edition, BPB Publisher, 2010.
3. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, –Operating System Concepts‖, 9th Edition, John Wiley & Sons, 2015.
4. William Stalling, –Operating Systems: Internals and Design Principles‖, 9th edition, PHI, 2018.
5. Behrouz A. Forouzan, Richard F. Gilbery, –Unix and shell Programming‖, 1st Edition, Cengage Learning India, 2003.

WEB REFERENCES:

1. https://nptel.ac.in/content/storage2/courses/106108101/pdf/PPTs/Mod_13.pdf
2. https://nptel.ac.in/courses/117106113/

Prepared By: Dr.K.S.Deepthi, Mr.K.Chandra Sekhar