ANNEXURE - II

R-20SECOND YEAR SYLLABUS

SECOND YEAR SEMESTER – I												
Code	**Course**	**Category**	**L**	**T**	**P**	**E**	**O**	**Total**	**Sessional Marks**	**External Marks**	**Total Marks**	**Credits**
IT211	Data Structures	PC	2	1	0	1	4	8	40	60	100	3
IT212	Basics of Electrical Engineering	ES	3	0	0	1	4	8	40	60	100	3
IT213	Discrete Mathematical Structures	BS	3	0	0	1	5	9	40	60	100	3
IT214	Software Engineering	PC	3	0	0	1	2	**6**	40	60	100	3
IT215	Computer Organization and Microprocessors	PC	2	1	0	1	3	7	40	60	100	3
IT216	Data Structures Lab	PC	0	0	3	0	3	6	50	50	100	1.5
IT217	Microprocessors Lab	PC	0	0	3	0	3	6	50	50	100	1.5
IT218	Python Programming Lab	SC	0	1	3	0	1	5	50	50	100	2.5
IT219	Constitution of Indian & -Intellectual Property Rights	MC	3	0	0	0	1	4	50	0	50	
TOTAL			**13**	**3**	**9**	**5**	**25**	55	400	450	**850**	**20.5**

SECOND YEAR SEMESTER – II												
Code	**Course**	**Category**	**L**	**T**	**P**	**E**	**O**	**Total**	**Sessional Marks**	**External Marks**	**Total Marks**	**Credits**
IT221	Operating Systems	PC	2	1	0	1	3	7	40	60	100	3
IT222	Probability Statistics & Queuing Theory	BS	3	0	0	1	6	10	40	60	100	3
IT223	Data Communications	PC	3	0	0	1	2	6	40	60	100	3
IT224	Database Management Systems	PC	2	1	0	1	3	7	40	60	100	3
IT225	Object Oriented Programming using JAVA	PC	3	0	0	1	4	8	40	60	100	3
IT226	Object Oriented Programming using JAVA Lab	SC	0	0	3	0	3	6	50	50	100	1.5
IT227	Operating Systems Lab	PC	0	0	3	0	3	6	50	50	100	1.5
IT228	Database Management Systems Lab	PC	0	0	3	0	3	6	50	50	100	1.5
TOTAL			**13**	**2**	**9**	**5**	**27**	56	350	450	**800**	**19.5**

SEMESTER-1

DATA STRUCTURES

COURSE CODEIT211 **CREDITS 3**

L T P E O
2 1 0 1 4

Sessional Marks: 40
End Exam Marks: 60
End Exam:3 Hours

Prerequisite:CProgramming

CourseObjective:

- Assess how the choice of data structures impacts the performance of programs.
- Choose the appropriate data structure and algorithm design method for a specified application.
- Solve problems using data structures such as linear lists, stacks, queues, hash tables, binary trees, binary search trees and graphs.

Course Outcomes: After completion of this course student will be able to:

CO-1: Compare and contrast static and dynamic implementation of data structures : stacks, queues.

CO-2: Select appropriate Searching and sorting technique for a given dataset

CO-3: Design and implement abstract data types such as linked list, stack, queue and tree in static and dynamic context using C programming language.

CO-4: Solve problems involving non-linear data structures: graphs, trees

Mapping of Course Outcomes with POs and PSOs

COs/ POs-PSOs	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
CO1	2	3	3	1	1			1	1	1		1	3	11
CO2	2	2	3		1			1	1	1		1	3	1
CO3	2	2	3		1			1	1	1		1	3	1
CO4	2	2	3	3	1			2	3	1		1	3	2

SYLLABUS

Unit-I: Introduction **10 Periods**

Introduction to data structures, arrays and structures. Dynamic Memory Management, AbstractData Type (ADT).

List: Definition and examples- Primitive Operations- Representation using array and Linked List. Types of Linked Lists and implementation: single, double and circular. The array and linked list advantages, disadvantages and applications.

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

- Implement List data structure using array and Linked list.
- Compare advantages & disadvantages of static and dynamic implementations.

Unit-II: Stacks and Queues **12 Periods**

The Stack ADT: Definition, Primitive Operations and representation. Stack ADT implementation using array and linked list. Applications of Stacks: Prefix, infix and Postfix notations, conversion between infix, prefix and postfix, postfix evaluation using stacks.

Queue ADT: Definition, Primitive operations and Representation. Queue ADT implementation using array and linked list. Types of Queue: Circular Queue, Priority Queue, Operations and implementation using array and linked list. Applications of Queues.

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

- Describe specific problems to which stacks and queues are suited.
- Demonstrate the operations of stacks and queues.
- Apply stacks to a specific application.
- Apply queues to a specific application.

Unit - III: Sorting and Searching **10 Periods**

Sorting: General background, bubble sort, insertion sort, quick sort and merge Sort.

Searching: General background, linear search, binary search.

Introduction to Hashing, Hash Function, Hashing techniques, Collision Resolution Methods: Open Addressing, Chaining.

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

- Understand common searching and sortingtechniques.
- Compare quick, and merge sorting in terms of their overall runtime efficiency.
- Chart the efficiency of quick, and merge sorting for small, medium, and large data.

Unit-IV: Trees **8 Periods**

Trees: Introduction, Terminology, Binary trees: Terminology, Representation. Binary tree implementation using array and linked list. Tree Traversal Techniques,

Types: Binary Search Tree, AVL Tree.

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

- Know the difference between binary search trees and AVL Trees.
- Apply trees to solve specific application requirements.

Unit-V: Graphs **12 periods**

Graphs: Introduction- terminology, Representation of graphs-Adjacency list and adjacency matrix, Representation in C, Warshall's Algorithm.

Graph Traversals-Breadth-First Search, Depth-First Search.

Spanning Trees: Introduction and terminology, Minimum Spanning Trees.algorithms: Primsand Krushkals. Applications of Graphs: Dijkstra's Algorithm for single source shortest path.

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

- Describe the two principal graph traversal paradigms.
- Demonstrate the use of graphs as a solution to a particular application requirement.
- Know the difference between directed and undirected graph.
- Explain means of generating spanning trees.

<u>Text Books:</u>

1. Y. Langsam, M. Augenstin and A. Tannenbaum, "Data Structures using C" PearsonEducation, 2nd Edition, 1995

<u>Reference Books:</u>

1. Ellis Horowitz, Sartaj Sahni, Fundamentals of Data Structure, computer science Press.
2. Richard F, Gilberg , Forouzan, Cengage ,"Data Structures", 2/e, 2005.

BASICS OF ELECTRICAL ENGINEERING

COURSE CODEIT211
L T P E O
30 0 1 4

CREDITS 3
Sessional Marks: 40
End Exam Marks: 60
End Exam:3 Hours

Prerequisites: Nil

Course Objectives:

- □ Analysis of circuits by using KCL and KVL
- □ Finding equivalent circuits by using circuit theorems
- □ Analysis of magnetic circuits
- □ Principle of operation and behavior of electrical machines

Course Outcomes:

By the end of the course, the student will be able to:	
1.	Calculate voltage a cross, current through and power supplied / absorbed by an electrical
2.	Analyze magnetic circuit
3.	Obtain the performance characteristics of D.C. Generators.
4.	Obtain the performance characteristics of D.C. Motors.
5.	Obtain the performance characteristics of Transformer and Induction Motor.

SYLLABUS

UNIT –I
Electric Circuits : Circuit Elements, Basic Law’s, KVL, KCL, Linearity Principle (Super Position), Mesh and Nodal analysis, Thevenin’s and Norton’s theorems.

UNIT-II
Magnetic Circuits : Definitions of magnetic circuit, Reluctance, Magneto-motive force, magnetic flux, Simple problems on magnetic circuits. Faraday’s laws of Electromagnetic Induction, Induced E.M.F., Dynamically induced E.M.F.

UNIT-III
D.C. Generators : D.C. Generator principle, construction of D.C. generator, E.M.F equation ofD.C. generator, Types of D.C. generators, Efficiency, Applications.

UNIT-IV

Motors : Principle, working of D.C. Motors, Significance of back E.M.F., Torque equation of D.C. Motors, Types of D.C. Motors, Special Motors (Stepper Motor and Servo Motor) and Applications.

UNIT-V

AC Machines : Transformer working Principle, EMF equation of transformer, Voltage regulation of Transformer. Three-phase Induction Motor working principle, Construction of 3
Phase Induction Motor, Principle of operation, Types of 3 phase induction Motors, Applications.

TEXT BOOKS:

1. **V.K. MEHTA &ROHIT MEHTA** *"Principles of Electrical Engineering"* 2nd edition, S. Chand Publications.
2. **V.K. MEHTA & ROHIT MEHTA** *"Principles of Electrical Machines"* 2nd edition, S. Chand Publications.

REFERENCE BOOK:

1. **J.B. Gupta** *"A Text book of Electrical Engineering"* S.K. Kataria& SonsPublications.

DISCRETE MATHEMATICAL STRUCTURES
[common to CSE & I.T.]

COURSE CODEIT211	**CREDITS 3**
L T P E O	Sessional Marks: 40
30 0 1 5	End Exam Marks: 60
	End Exam:3 Hours

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

Course Objectives :

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

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 ordered set.

(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.

SOFTWARE ENGINERRING

COURSE CODE –CATEGORY: IT214 **CREDITS 3**
L T P E O Sessional Marks: 40
3 0 0 1 2 End Exam: 3 Hours

End Exam Marks: 60

Pre requisites: computer fundamental, any programming languages

Course Objectives

- The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
- Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams.

Course Outcomes

1. Ability to translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
2. Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
3. Will have experience and/or awareness of testing problems and will be able to develop a simple testing report.
4. To manage time, processes and resources effectively by prioritising competing demands to achieve personal and team goals Identify and analyzes the common threats in each domain.

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	3	3	1	1	1				2	2	3	3
CO2	2	2	2	2	1					2	2	2	3	3
CO3	2	2	3	3	3						1	1	3	3
CO4	2	2	3	3	3						1	1	3	3

UNIT - I **8 Periods**

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

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

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

Learning outcomes:

1. Basic knowledge and understanding of the analysis and design of complex systems.
2. Ability to apply software engineering principles and techniques.

UNIT – II **10 Hours**

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

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

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

Learning outcomes:

1. defined as a process of analyzing user requirements
2. designing software application which will satisfy that requirements

UNIT – III **12 Hours**

Design Engineering: Design process and design quality, design concepts, the design model.

Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

Learning outcomes:

1. To produce efficient, reliable, robust and cost-effective software solutions.
2. Ability to perform independent research and analysis.

UNIT – IV **10 Hours**

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging.

Product metrics: Software quality, metrics for analysis model, metrics for design model, metrics for source code, metrics for testing, metrics for maintenance.

Learning outcomes:

1. check whether the actual software product matches expected requirements
2. making it efficient and effective as per the quality standards defined for software products

UNIT - V **10 Hours**

Metrics for Process and Products: Software measurement, metrics for software quality.

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM, RMMM plan.

Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards.

Learning outcome:

1. Ability to understand and meet ethical standards and legal responsibilities.

2. Ability to develop, maintain and evaluate large-scale software systems.

TEXT BOOKS:

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

REFERENCE BOOKS:

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

COMPUTER ORGANIZATION AND MICROPROCESSORS

COURSE CODE: IT215 **Credits:3**

L	T	P	E	O
2	1	0	1	3

SessionalMarks: 40

EndExam Marks: 60

EndExam:3Hours

Prerequisite:

Students are expected to be capable of recollecting the concepts of Fundamentals of IT and DLD to relate with the linked topics in CO

Course Objectives

1. Understand the fundamentals of different instruction set architectures and their relationship to the CPU design.
2. To study the different ways of communicating with I/O devices and standard I/O interfaces.
3. To study the hierarchical memory system including cache memories
4. To introduce students with the architecture and operation of typical microprocessors
5. Edit, compile, execute, and debug an assembly language programs with loops, interrupts and subroutines.

Course outcomes

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

CO-1: Identify and compare different issues related to organization of CPU, hardwired and micro programmed control unit

CO-2: Categorize memory organization, input-output organization of a computer and explain the function of each element related to them.

CO-3: Describe the architecture of microprocessors 8085/8086. Illustrate machine level instructions with timing diagrams

CO-4: Write debug and analyze assembly language programs for the 8085/8086 microprocessor instruction set.

Mapping of Course Outcomes with POs and PSOs

COs/POs -PSOs	PO1	PO2	PO 3	PO 4	PO 5	PO 6	PO 7	PO8	PO 9	P010	PO1 1	PO12	PSO 1	PSO 2
CO1	2	2	2	3			1	1	1			1	2	1
CO2	2	2		3			2	1	1		2	1	2	1
CO3	1	2						1	1	1		1	1	1
CO4	2	3			2			1	1	1		1	1	1

UNIT-1 **10 Periods**

Central Processing Unit: Introduction, General Register Organization, Stack Organization, Program Control, Program Interrupt, Types of interrupts, CISC Characteristics, RISC Characteristics.
Micro programmed Control: Control Memory, Address Sequencing, Micro program Example.

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

1. Understand concepts of Hardwired control and micro programmed control.
2. Understand the architecture and functionality of central processing unit.

UNIT-2 **14 Periods**

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, Cache Memory, virtual memory concept.
Learning Outcomes: At the end of this Unit the student will be able to

1. Discuss about different types of peripheral devices of computer.
2. Discuss the concept of memory organization.
3. Explain the use of cache memory.
4. Summarize the types of memory.

UNIT-3 8085 Microprocessor Architecture **10 Periods**

Internal Architecture and Functional/Signal Description of typical 8-bit µP, Instruction set, instruction types and formats; Instruction execution, instruction cycles, different types of machine cycles and timing diagram.

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

1. Describe the 8085 pin configuration and its working with the help of diagram.
2. Illustrate 8085 machine instructions
3. Illustrate programming techniques with example problems.

UNIT-4 Instruction Subroutines and Interrupts **12 Periods**

Types of operations and operands, encoding an instruction set, assembly language programming, addressing modes and formats, loops, subroutines,16-bit data Operations, Interrupts and Interrupt Service Routines .
Learning Outcomes: At the end of this Unit the student will be able to

1. Illustrate programming techniques with subroutines and interrupts.
2. Learn how to encode a program to AL.

UNIT-5 The 8086 µP. Architecture and Instruction Set **10 Periods**

Internal Architecture and Functional/Signal Description of 8086/8088. Segmented Memory, Maximum-Mode and Minimum-Mode Operation, Addressing Modes, Instruction Set.
Learning Outcomes: At the end of this Unit the student will be able to

1. Describe the internal architecture of 8086 microprocessor.
2. Illustrate the different modes of accessing memory with examples.
3. Describe the memory management schemes in 8086 microprocessor

TEXT BOOKS

1. Computer System Architecture, M.Morris Mano ,Third Edition, Pearson Education Inc., 2003.
2. Microprocessor Architecture, Programming, and Applications with the 8085 Ramesh S. Gaonkar , 4th Edition, Penram International, 1999
3. The 80x86 Family, Design, Programming and Interfacing, John E.Uffenbeck, 3rd Edition, Pearson Education Inc., 2002.

REFERENCES

1. John D. Carpinelli , Computer Systems Organization and Architecture, ,Pearson Education Inc., 2003
2. William Stallings, Computer Organization and Architecture, 5th Edition,2000.
3. Microprocessors and Interfacing, Douglass V Hall, 2nd Edition, TMH Publishing.

CHANGE OF SYLLABUS

1. New theory subject formed by combining computer organization and Microprocessor. 2 units from CO and 3 units from MP are included.

Note: This course is related to Employability/Skill development.

DATA STRUCTURES LAB

COURSE CODEIT216
L T P E O
0 0 3 0 3

CREDITS 1.5
Sessional Marks: 50
End Exam Marks: 50
End Exam:3 Hours

Prerequisite:
C Programming, Data Structures.

Course Objectives:

- Assess how the choice of data structures impacts the performance of programs
- Choose the appropriate data structure and algorithm design method for a specified application
- Solve problems using data structures such as linear lists, stacks, queues, hash tables, binary trees, heaps, binary search trees, and graphs and writing programs for these solutions

Course Outcomes: After completion of this course student will be able to:

CO-1: Compare performance of searching and sorting algorithms.
CO-2: Develop Programs by employing static and dynamic memory management.
CO-3: Apply suitable data structure and algorithm to solve for the given data structure related problem.

Mapping of Course Outcomes with POs and PSOs:

COs/POs-PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	2	3	3	1			1	3	1		1	3	2
CO2	1	2	3	3	1			1	3	1		1	3	2
CO3	1	2	3	1	1			2	3	1		1	3	

Note: Every lab must be practiced in GDB Compiler/HackerRank platform and the execution part of rubrics (apart from viva, observation and record) must be evaluated based on the GDB/HackerRank performance.

List of Programs:

1) Programs to implement the following using an array. [CO1]
a) Stack b) Queue

2) Programs to implement the following using a singly linked list. [CO1]
a) Stack b) Queue

3) Program to do the following [CO1]
a) Infix to postfix conversion. b) Evaluation of postfix expression.

4) Programs to implement the following data structures. [CO1]
a) Circular Queue b) Priority Queue

5) Program to perform the following operations: [CO2]
a) Insert an element into a binary search tree.
b) Delete an element from a binary search tree.
c) Search for a key element in a binary search tree.

6) Program that use non-recursive functions to traverse the given binary tree in a) Preorder
b) In-order c) Post-order. [CO2]

7) Program to implement BFS and DFS for a given graph. [CO2]

8) Program to implement the following sorting methods: [CO3]
a) Merge sort b) Quick sort c) Insertion Sort d) Bubble Sort

9) Program to implement the following searching methods: [CO3]
a) Linear Search b) Binary search

10) 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, where m is size of the hashtable. [CO3]

Reference Books:

1. Y. Langsam, M. Augenstin and A. Tannenbaum, “Data Structures using C” Pearson Education, 2nd Edition, 1995.
2. Richard F, Gilberg, Forouzan, Cengage, Data Structures, 2/e, 2005.
3. Data Structures using C, 2/2, ISRD Group.

MICROPROCESSOR LAB

COURSE CODE:IT217 Credits: 1.5
L T P E O Sessional Marks: 50
0 0 3 0 3 End Exam Marks: 50
End Exam: 3Hours

Course Objectives

1. Ability to write Assembly Language programs for the Intel 8085/8086 microprocessors using trainer kits and MASM. Ability to load, verify, and save machine language programs.
2. Ability to debug and interpret machine code using the DEBUG software.
3. Ability to decode and encode machine code by hand.
4. Ability to examine and modify the contents of Memory.

Course Outcomes

1. Design digital logic circuits.
2. Write debug and analyse assembly language programs for the 8085/8086 microprocessor instruction set.
3. Execute assembly language programs using trainer kits/TASM or MASM software and analyze and interpret the results.

Mapping of Course Outcomes with POs and PSOs

COs/POs-PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	P010	PO11	PO12	PSO1	PSO2
CO1	1	2	3		2			1	1	1		1	1	1
CO2	1	3	1	1	2			1	1	1		1	1	1
CO3	2	3	1	1	2			1	1	1		1	11	1

Digital Logic Design (DLD) Experiments **CO1**

1. Characteristics of TTL
   1.1 AND gate
   1.2 OR gate
   1.3 NOT gate
   1.4 NAND gate
   1.5 NOR gate
   1.6 XOR gate
2. Half adder
3. Full adder
4. Multiplexer
5. Jk flip-flop
6. BCD to 7 segment decoder
7. Universal shift register
8. Decode counter

Microprocessor 8085 experiments **CO2**

9. ALP to perform addition of two 8-bit numbers
10. ALP to perform subtraction of two 8-bit numbers
11. ALP to perform multiplication of two 8-bit numbers
12. ALP to perform division of two 8-bit numbers
13. ALP to find largest number in array
14. ALP to find smallest number in array
15. ALP to arrange an array in ascending order
16. ALP to arrange an array in Descending order
17. Logical operations
18. ALP to perform 16- bit addition
19. ALP to perform 16-bit subtraction
20. ALP to perform addition of two unsigned BCD numbers
21. ALP to perform different multiplication tables

Microprocessor 8086 experiments (programs are implemented using TASM (turbo assembler) CO-3

22. ALP to perform addition of two 8-bit numbers
23. ALP to perform subtraction of two 8-bit numbers
24. ALP to perform multiplication of two 8-bit numbers
25. ALP to perform division of two 8-bit numbers
26. ALP to perform addition of two 16-bit numbers
27. ALP to perform subtraction of two 16-bit numbers
28. ALP to perform multiplication of two 16-bit numbers
29. ALP to perform division of two 16-bit numbers
30. ALP to perform swapping of two 16-bit data
31. ALP to perform addition of even numbers and odd numbers separately from an array of data
32. ALP to add elements in an array
33. ALP to find whether the given number is even or odd
34. ALP to perform factorial of a number
35. ALP to copy a string from one location to another location
36. ALP to perform reverse of string operation
37. ALP to find largest number in an array
38. ALP to find smallest number in an array
39. ALP to search a particular character in a string
40. ALP to count no of zeros and ones in a byte of data

PYTHON PROGRAMMING LAB

COURSE CODE IT218 **CREDITS 2.5**

L T P E O
0 1 3 0 1

Sessional Marks: 50
End Exam Marks: 50
End Exam: 3 Hours

Prerequisite: C Programming

Prerequisite:
Students are expected to be able to open command prompt window or terminal window, edit a text file, download and install software, and understand basic programming concepts.

Course Objectives:

- To develop Python programs with conditional statements , loops and functions
- To use Python data structures – dictionaries, tuples and sets
- To do input/output with files in Python.
- To develop Python programs using OOP concepts.

Course Outcomes: After completion of this course student will be able to:

CO-1: Apply the fundamental concepts to explore the methods of mutable and immutable data structures.

CO-2: Explore the built-in functions of regular expressions and file systems.
CO-3: Use/Utilize the principles of Object oriented programming in developing Python scripts.

Mapping of Course Outcomes with POs and PSOs:

COs/ POs-PSOs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PS O1	PS O2
CO1	3	3	3	3	2	2			2		1	2	3	2
CO2	3	3	2	3	3	3			3		2	3	3	3
CO3	3	3	3	3	3	3			3		1	3	3	3

UNIT-I

Parts of Python Programming Language: Identifiers, Keywords, Statements and Expressions, Variables, Operators, Precedence and Associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversions, The type() Function and Is Operator, Dynamic and Strongly Typed Language.

Control Flow Statements, The if, if…else, if…elif…else Decision Control Statements, Nested if Statement, The while, for Loops, The continue and break Statements.

Functions, Built-In Functions, Commonly Used Modules, Function
Definition and Calling the Function, The return Statement and void Function, Scope and Lifetime of
Variables, Default Parameters, Keyword Arguments, *args and **kwargs, Command Line Arguments.

Learning Outcome:

At the end of this Unit the student will be able to

- Analyze fundamental advantages of python over the other programming languages.
- Solve, test and debug basic problems using python script
- Implement Flow control statements required to real world problems.
- Familiarize the usage of functions and Modules and packages to enhance the problem solving.

UNIT-II

Strings- Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings.

Lists- Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods, The del Statement.

Learning Outcome:

At the end of this Unit the student will be able to

- Manipulate python programs by using python data structures like lists and strings.

UNIT-III

Dictionaries-Creating Dictionary, Accessing and Modifying key:value Pairs in Dictionaries, Built- In Functions Used on Dictionaries, Dictionary Methods, The del Statement.

Tuples and Sets- Creating Tuples, Basic Tuple Operations, Indexing and Slicing in Tuples, Built-In Functions Used on Tuples, Relation between Tuples and Lists, Relation between Tuples and Dictionaries, Tuple Methods, Using zip() Function, Sets, Set Methods, Traversing of Sets.

Learning Outcome:

At the end of this Unit the student will be able to

- Manipulate python programs by using python data structures like dictionaries, tuples and sets.

UNIT-IV

Files-Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files.

Regular Expression Operations- Using Special Characters, Regular Expression Methods, Named
Groups in Python Regular Expressions.

Learning Outcome:

At the end of this Unit the student will be able to

- Design python programs using File operations.
- Design python programs using Regular Expression Methods.

UNIT-V

Object-Oriented Programming- Classes and Objects, Creating Classes in Python, Creating Objects in Python, The Constructor Method, Classes with Multiple Objects, Class Attributes versus Data Attributes.

Encapsulation, Inheritance and Polymorphism.

Learning Outcome:

At the end of this Unit the student will be able to

- Design object-oriented programs with Python classes.
- Usage of inheritance, encapsulation, inheritance and polymorphism for reusability

Textbooks:

1. **"Introduction to Python Programming",** by Gowrishankar S, Veena A, 1st Edition, CRC Press/Taylor & Francis, 2018. ISBN-13: 978-0815394372

Reference Textbook:

1. Programming python, by Mark Lutz, 4th Edition, O'REILLY
2. The complete reference python, by Martin C Brown

Sample list of Experiments:

1. A) Running instructions in Interactive interpreter and a Python Script. [CO1]

   B) Write a program add.py that takes 2 numbers as command line arguments and prints its sum. [CO1]

2. A) Write a program to compute distance between two points taking input from the user. [CO1]

   B) Write a Program for checking whether the given number is a even number or not. [CO1]

3. A) Write a program to find greatest among 3 numbers.

   B) Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, . . . , 1/10. [CO1]

4. A) Write a program using a while loop that asks the user for a number, and prints a count down from that number to zero. [CO1]

   B) Write a program using for loop to print sum of prime numbers up to the given number. [CO1]

5. A) Write a program using for loop to print sum of even terms in a Fibonacci series up to the given number. [CO1]

   B) Write a program using for loop to find the sum of cosine series up to specified number of terms for a given value of X. [CO1]

6. A) Write a program to count the number of characters in the string and store them in a dictionary data structure. [CO2]

   B) Write a program to use split and join methods in the string and trace a given word in the sub strings. [CO1]

7. A) Write a program to combine lists. Combine these lists into a dictionary as well. [CO3]
   B) Find mean, median, mode for the given set of numbers in a list. [CO3]

8. A) Write a program to print each line of a file in reverse order. [CO3]
   B) Write a program to compute the number of characters, words and lines in a file. [CO3]

9. A) Write function balls collide that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding. [CO1]

Hint: Represent a ball on a plane as a tuple of (x, y, r), r being the radius

If (distance between two balls centers) <= (sum of their radius) then (they are colliding).

B) Write a function Reverse to reverse a list. Without using the reverse function. [CO1]

10. A) Write function to compute gcd, lcm of two numbers. Each function shouldn't exceed one line. [CO1]

    B) Write function Unique to check uniqueness of two lists. Function should return a Boolean true if they are unique otherwise return false. [CO1]

11. A) Write a program to perform addition of two square matrices. [CO1]
    B) Write a program to perform multiplication of two square matrices. [CO1]

12. A) Write a program to perform different operations on sets. [CO2]
    B) Write a program to perform different operations on tuples. [CO2]

13. Write a program to perform compile(), findall(), split(), sub() , subn() functions on expressions using re. [CO3]

14. A) Write a program to illustrate concept of multi level inheritance. [CO4]
    B) Write a program to illustrate concept of multiple inheritance. [CO4]

15. A) Write a program to illustrate concept of method overloading. [CO4]
    B) Write a program to illustrate concept of method overriding. [CO4]

CONSTITUTION OF INDIA & INTELLECTUAL PROPERTY RIGHTS
(CI-3 & IPR -2)

IT ***Credits: NILL***

L	T	P	E	O	
2	1	0	0	0	

Sessional Marks: 50
End Exam Marks: NILL

Course Objectives

- To impart knowledge in basic concepts of Constitution of India
- To understand the fundamental principles of Intellectual Property Rights and its importance

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

	COURSE OUTCOMES
CO-1	To impart basic knowledge about the Constitution of India
CO-2	Comprehend the Fundamental Rights and Fundamental Duties of the Indian Citizen to implant morality, social values and their social responsibilities.
CO-3	Familiarize with distribution of powers and functions of Local Self Government, state and central policies and amendment procedure
CO-4	Understand the fundamental principles of IPR
CO-5	Appraise of IP rights like patents, industrial design, trademark, copyrights for effective protection and utilization of their innovations.

Module 1-Introduction and Basic Information about Indian Constitution:
Meaning of the constitution law and constitutionalism, Historical Background of the Constituent Assembly, Government of India Act of 1935 and Indian Independence Act of 1947 , Enforcement of the Constitution, Indian Constitution and its Salient Features, Preamble of the Constitution.

Module 2 - Fundamental Rights and Directive Principles
Scheme of Fundamental Rights, Fundamental Duties, Directive Principles of State Policy – Its importance and implementation, Scheme of the Fundamental Right to certain Freedom under Article 19, Scope of the Right to Life and Personal Liberty under Article 21

Module 3 - Administrative organisation&**Amendments**
Federal structure and distribution of legislative and financial powers between the Union and the States, Parliamentary Form of Government in India – The constitution powers and status of the President of India, Amendment of the Constitutional Powers and Procedure,

Module 4 - Intellectual Property Rights Information:
Introduction to IPRs, Basic concepts and need for Intellectual Property – Patents, Copyrights, Geographical Indications, IPR in India and Abroad – Genesis and Development – the way from WTO to WIPO –TRIPS, Nature of Intellectual Property, Industrial Property, technological Research, Inventions and Innovations – Important examples of IPR

Module 5 - REGISTRATION OF IPRs
Meaning and practical aspects of registration of Copy Rights, Trademarks, Patents, Geographical Indications, Trade Secrets and Industrial Design registration in India and Abroad

Text Books:

1.V.Scople Vinod, Managing Intellectual Property, Prentice Hall of India pvt Ltd, 2012

2. S. V. Satakar, —Intellectual Property Rights and Copy Rights, Ess Publications, New Delhi, 2002

3. Brij Kishore Sharma: *Introduction to the Indian Constitution*, 8th Edition, PHI LearningPvt. Ltd.

4. Granville Austin: *The Indian Constitution: Cornerstone of a Nation (Classic Reissue)*Oxford University Press.

References:

1. Deborah E. Bouchoux, —Intellectual Property: The Law of Trademarks, Copyrights, Patents and Trade Secrets, Cengage Learning, Third Edition, 2012.

2. Prabuddha Ganguli,Intellectual Property Rights: Unleashing the Knowledge Economy, Mc Graw Hill Education, 2011.

3. Edited by Derek Bosworth and Elizabeth Webster, The Management of IntellectualProperty, Edward Elgar Publishing Ltd., 2013.

4. Subhash C. Kashyap: *Our Constitution: An Introduction to India's Constitution and constitutional Law*, NBT, 2018.

5. Madhav Khosla: *The Indian Constitution*, Oxford University Press.

PM Bakshi: *The Constitution of India*, Latest Edition, Universal Law Publishing.

SEMESTER-2

OPERATING SYSTEMS

COURSE CODEIT211 **CREDITS 3**

L T P E O
2 1 0 1 4

Sessional Marks: 40
End Exam Marks: 60
End Exam:3 Hours

Prerequisite:

Knowledge in Computer Organization.

Course Objectives:

□ Understand Functions, Services and structure of Operating Systems.
□ Understand processes, threads, schedulers and explanation of CPU
□ scheduling.
Understand issues related to Process Synchronization and focus on principles of Deadlock and related problems
□ Comprehend the mechanisms used in Memory Management and Virtual Memory.
□ Understand the concepts of File System, secondary storage management and Disk Scheduling

Course Outcomes:

1. Illustrate the overall view of operating system structure and process management.
2. Apply appropriate synchronization techniques in handling deadlocks.
3. Interpret the issues and challenges of memory management
4. Demonstrate the concepts of file system implementation and protecting mechanisms.

Mapping of course outcomes with program outcomes:

COs/POs-PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	P010	PO11	PO12	PSO1	PSO2
CO1	2	1	2	3	1					1	2	3	3	3
CO2	3	1	2	2				2	2		1	1	1	3
CO3	3	2	2	1	2			3	2		1	2	2	3
CO4	2	2	1	1	2		1	2	1			2	2	3

UNIT – I 10 Periods

INTRODUCTION TO OS AND PROCESS MANAGEMENT

Introduction to operating systems ,operating system structures ,system calls, Process concept, CPU Scheduling: Scheduling criteria ,Scheduling algorithms , Multiple processor scheduling ,Real time scheduling ,Algorithm Evaluation. ,Operations on processes ,Cooperating processes ,Interprocess communication. Multi threaded programming.

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

1. Explain Types of operating systems
2. Describe process states and process models
3. Compare processor scheduling algorithm

UNIT – II 10 Periods

PROCESS SYNCHRONIZATION AND DEADLOCK

Process Synchronization: The critical section problem ,Synchronization hardware ,Semaphores , Classic problems of synchronization ,critical regions , Monitors.

Deadlock: System model ,Deadlock characterization ,Methods for handling deadlocks ,Deadlock prevention , Deadlock avoidance, Deadlock detection ,Recovery from deadlock.

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

1. Describe race condition & mutual exclusion
2. Identify Deadlocks
3. Apply Deadlock recovery procedure

UNIT – III 10 Periods

MEMORY MANAGEMENT

Memory Management: Background – Swapping – Contiguous memory allocation – Paging – Segmentation – Segmentation with paging. Virtual Memory: Background – Demand paging – Process creation – Page replacement – Allocation of frames – Thrashing.

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

1. Describe memory management
2. Differentiate Contiguous and Non contiguous memory
3. Differentiate physical and virtual primary memory

UNIT – IV 8 Periods

FILE SYSTEMS AND ITS IMPLEMENTATION

File System Interface: File concept – Access methods – Directory structure – File system mounting – Protection. File System Implementation : Directory implementation – Allocation methods – Free space management – efficiency and performance – recovery – log structured file systems.

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

1. Apply file management concepts in Operating System
2. Explain Directory structure of Operating System

UNIT – V 10 Periods

SECONDARY STORAGE STRUCTURES AND PROTECTION

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, Capability Based systems.

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

1. Describe Disk organization
2. Implement file system security

CASE STUDY (Not considered in the examination): THE LINUX OPERATING SYSTEM: Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory management; File systems, Input and output; Inter process communication

Text Book:

1. Silberschatz, Galvin, and Gagne, "Operating System Concepts", Sixth Edition, Wiley India Pvt Ltd, 2003.

Reference Books:

1. Andrew S. Tanenbaum, "Modern Operating Systems", Second Edition, Pearson Education, 2004.
2. Gary Nutt, "Operating Systems", Third Edition, Pearson Education, 2004.
3. Harvey M. Deitel, "Operating Systems", Third Edition, Pearson Education, 2004.

PROBABILITY, STATISTICS AND QUEUING THEORY

[Common to CSE& I.T.]

COURSE CODEIT211	**CREDITS 3**
L T P E O	Sessional Marks: 40
2 1 0 1 4	End Exam Marks: 60
	End Exam:3 Hours

Prerequisites: Elementary knowledge of Set theory, Combinations , Calculus and basic Statistics .

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.

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.

Mapping of course outcomes with program outcomes:

COs/POs-PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	P010	PO11	PO12	PSO1	PSO2
CO1	3	3									3		3	3
CO2	2	3									3		2	3
CO3	3	3									3		3	3
CO4	3	3									3		2	2
CO5	3	3									3		2	2

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 (L3)
- Explain the concept of a random variable and the probability distributions (L5)
- Express the features of discrete and continuous random variables and explain about probability mass, density function and formulate the distribution functions. (L5)
- Calculate the expected value of a random variable and moments and formulates the Moment Generating Function (L3)

UNIT- II:Probability Distributions (14 Periods)

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

Continuous ProbabilityDistributions: UniformDistribution,ExponentialDistribution, Memorylessproperty,NormalDistribution,propertiesofNormalDistribution,Importanceof 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 (L2)
- Understand importance of continuous distributions Exponential, Uniform and Normal and Exponential Distribution and solve the problems about these distributions(L2)
- calculate probabilities of events for these distributions using the probability function, probability density function or cumulative distribution function (L3)

UNIT - III: Curve Fitting , Correlation and Regression (10 Periods)

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

Correlation: Definition, Karl Pearson's Coefficient of CorrelationMeasures 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(L2)
- Calculate Pearson's correlation coefficient, Spearman's rank correlation coefficient (L3)
- Compute and interpret simple linear regression between two variables (L3)
- Calculate regression coefficients and study the properties of regression lines (L3)

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 (L6)
- 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 (L6)
- Perform and analyze hypotheses tests of means, proportions and variances using bothone- and two-sample data sets (L4)
- Apply the appropriate Chi-Squared test for independence and goodness of fit (L3)

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

DATA COMMUNICATIONS

COURSE CODEIT211
L T P E O
2 1 0 1 4

CREDITS 3
Sessional Marks: 40
End Exam Marks: 60
End Exam:3 Hours

Course Objectives:

- ❑ Understand the basics of data communication and networking concepts included the OSIreference model
- ❑ List the physical layer protocols and their roles
- ❑ Describe physical layer signaling and encoding
- ❑ Explain how electrical signals can be used to represent data bits
- ❑ Understand basic models of communication with the help of telephone data communication

Course Outcomes:

CO1: Describe the basic concepts of Data Communication, network models and their applications.

CO2: Illustrate Analog, Digital modulation techniques and Multiplexing techniques.

CO3: Classify transmission medias and switching techniques for effective reliable communication.

CO4: Demonstrate telephone and cable networks for data transfer in communication model.

Mapping of course outcomes with program outcomes:

COs/POs-PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	P010	PO11	PO12	PSO1	PSO2
CO1	3	2	2									2	2	2
CO2	2	2	2									2	2	2
CO3	1	2	2			2						2	2	2
CO4	2	2	2									2	1	1

SYLLABUS

UNIT-I **(8 hours)**

Overview: Introduction-Data Communications, Network, the Internet, Protocols and Standards. **Network Models**- Layered tasks, The OSI Model, Layers in the OSI Model, TCP/IP Protocol Suite, and Addressing.

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

- Understand the rudiments of how computers communicate
- Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.
- Explain the addressing and naming schemes used in network communications

UNIT-II **(8 hours)**

PhysicalLayerandMedia: **Data and Signals**- Analog and Digital, Periodic Analog Signals, Digital Signals, Transmission Impairment, Data Rate Limits, Performance. **Digital Transmission**-Digital-To-Digital Conversion, Analog-To-Digital Conversion, Transmission Modes.

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

- Describe physical layer signaling and encoding of data in handling Analog and Digital signals for communication through a channel.
- Explain how electrical signals can be used to represent data bits

UNIT-III **(8 hours)**

Analog Transmission-Digital-To-Analog Conversion, Analog-To-Analog Conversion. **BandwidthUtilization**: **Multiplexing and Spreading**-Multiplexing, Spread pectrum

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

- Understand data conversion schemes for effective communication through a channel.
- Enumerate how number of signals can be transmitted through a single physical medium at a time reducing the cost of transmission.
- Familiarize how bandwidth can be managed for effective communication

UNIT-IV **(7 hours)**

Transmission Media- Guided Media, Unguided Media: Wireless. **Switching-** Circuit-Switched Networks, Datagram Networks.

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

- List the basic characteristics of copper cable, fiber optic cable, and wireless network media
- **Analyze the network connectivity** for making one-to-one communication by switching the data from node to node through different switching techniques

UNIT-V **(7 hours)**

Switching -Virtual-Circuit Networks, Structure Of a switch. Using Telephone and Cable Networks for Data Transmission- Telephone Network, Dial-Up Modems, Digital Subscriber Line, Cable TV Networks, Cable TV For Data Transfer

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

- Understand the role of Virtual LANs (VLANs) in a switched LAN.
- Analyze the application aspects of data communication through telephone and cable networks.

Textbook:

1. Data Communications and Networking, Fourth Edition by Behrouza A. Forouzan,TMH.

Reference book:

1. Computer Networks, A.S.Tanenbaum,4th edition, Pearson education.
2. IntroductiontoDatacommunicationsandNetworking,W.Tomasi,Pearsoneducation.
3. Data and Computer Communications, G.S.Hura and M.Singhal, CRC Press, Taylor and FrancisGroup.

DATABASE MANAGEMENT SYSTEMS

COURSE CODEIT211 **CREDITS 3**
L T P E O Sessional Marks: 40
2 1 0 1 4 End Exam Marks: 60
End Exam:3 Hours

Prerequisite(s): Relational Algebra, Set Theory, knowledge in any program language
Course Objectives:

- □ Understand basic database concepts, including the structure and operation ofthe relational data model.
- □ Construct simple and moderately advanced database queries using StructuredQuery Language(SQL).
- □ Understand and successfully apply logical database design principles, includingE-R diagrams and databasenormalization.
- □ Understand the concept of a database transaction and related database facilities,including concurrency control, locking andprotocols.

Course Outcomes:
After completion of this course, the students will be able to:
CO1: ModelapplicationsdatarequirementsusingconceptualmodellingtoolslikeERdiagramsand design database schemas based on the conceptual model.
CO-2: Apply relational database theory and describe relational algebra expression, tuple and domain relation expression for queries.
CO-3: Write SQL commands to create tables and indexes, insert/update/delete data and query data in a relational DBMS. Optimize the database design by applying functional dependency and normalization principles
CO-4:. Examine the serializability of non-serial schedules and compare and contrast the concurrency control protocols.

Mapping of Course Outcomes with POs and PSOs

COs/POs-PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	P010	PO11	PO12	PSO1	PSO2
CO1	3	2	1	1	3				2	2		2	2	1
CO2	3	2	1	1	3				2	2		2	2	1
CO3	3	2	1	1	3				2	2		2	2	1
CO4	3	2	1	1	3				2	2		2	2	1

UNIT-I Introduction **10 Periods**

Introduction to DBMS: Overview, File system vs. DBMS, Advantages of DBMS, Structure of DBMS, Levels of Data Abstraction, Database Users and Administrators, E-R model: Entities, Attributes and Entity sets, Relationship and Relationship sets, Features of ER model, Conceptual database design with ER model.

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

- Understand database concepts and structures and querylanguage
- Understand the E Rmodel
- Design ER-models to represent simple database applicationscenarios

UNIT-II **12 Periods**

Relational model: Integrity constraints over relations and enforcement, Querying relation data, Logical database design, views, destroying/altering tables and views. Relational Languages: algebra and calculus.

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

- Understand the relationalmodel
- Convert the ER-model to relational tables, populate relational database andformulate SQL queries ondata
- Explain the basic concepts of relational model, relational database design,relational algebra and relationalCalculus

UNIT-III **10 Periods**

SQL: Basic SQL, Query, union, interest, except, Nested Queries, Aggregated Operation,
Null values, Embedded SQL, cursors, Database connectivity(ODBC and JDBC), Triggers and Active database, designing active databases.

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

- Execute various advance SQL queries
- Write SQL commands to create tables and indexes, insert/update/delete data, andquery data in a relationalDBMS.
- Perform PL/SQL programming using concept of Cursor Management, ErrorHandling, Packages andTriggers

UNIT-IV **12 Periods**

Normalization: Introduction to Schema Refinement - Problems Caused By Redundancy, Decomposition, Functional Dependency, Closure of a Set of FDs,Normal Forms (First, Second, Third normal forms, BCNF, Fourth & Fifth normalforms).

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

- Understand Functional Dependency and FunctionalDecomposition.
- Apply various Normalization techniques.
- Improve the database design bynormalization.

UNIT-V **10 Periods**

Transaction management: Transaction concept, transactions and schedules, concurrent execution of transactions Concurrency control: Lock management, specialized locking techniques, concurrency control without locking.

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

- understand transactions and their properties(ACID)
- understand the anomalies that occur withoutACID
- understand the locking protocols used to ensureIsolation

Text Books

1. Raghu Ramakrishnanand Johannes Gehrke, "Database Management Systems", 3rdEdition, McGraw-Hill,2003.

References Books

1. Silberschatz, Korth and Sudharshan, "Data Base System Concepts",5 Edition, McGraw Hill,2006.
2. Elmasri,Navathe,"FundamentalsofDatabaseSystems",5thEdition,PearsonEducation, 2007.

OBJECT ORIENTED OBJECT ORIENTED PROGRAMMING THROUGH JAVA

COURSE CODEIT225 **CREDITS 3**
L T P E O Sessional Marks: 40
3 0 0 14 End Exam Marks:60
End Exam:3 Hours

Prerequisite(s):

Basic knowledge on introduction to programming & object oriented programming concepts are essential.

Course Objectives:

- To understand object oriented programming concepts, and apply them in problem solving.
- To introduce the concepts of exception handling and multithreading.
- To introduce the design of Graphical User Interface using applets and swing controls.

<u>Course Outcomes</u>: After completion of this course student will be able to:

CO-1: Apply the OOPs concept s in writing simple java programs
CO-2: Develop programs for multithreading and exception handing to create new applications.
C0-3: Demonstrate the concepts of Java Files, collections and database in real-time problem solving.
C0-4: Develop GUI applications.

Mapping of course outcomes with program outcomes:

COs/POs-PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	1	3									3	3	2
CO2	2	2	3		2		2	2	2			2	3	2
CO3	2	2	3								2	2	3	2
CO4	2	2	3		2		2	2	2	2	2	2	3	3

SYLLABUS

UNIT-I: **(12 Periods)**

Classes and objects : creating classes and objects , accessing methods , object initialization , java garbage collector.
Inheritance - Inheritance hierarchies, super and sub classes, Member access rules, super keyword, preventing inheritance: final classes and methods, the Object class and its methods.
Polymorphism - dynamic binding, method overriding, abstract classes and methods.

Interfaces - Interfaces vs. Abstract classes, defining an interface, implementing interfaces, accessing implementations through interface references, extending interfaces.
Inner classes - uses of inner classes, local inner classes, anonymous inner classes, static inner classes, examples.
Packages - Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages.

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

- Define Objects and Classes and methods
- Describe Inheritance and method overriding
- Explain inheritance on interfaces, implementing interface, multiple inheritance using interface
- Create and understand package, importing package, access rules for packages, class hiding rules in a package

UNIT-II: (10 Periods)
Exception handling - Dealing with errors, benefits of exception handling, the classification of exceptions- exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, re throwing exceptions, exception specification, built in exceptions, creating own exception sub classes.
Multithreading - Difference between multiple processes and multiple threads, thread states, creating threads, interrupting threads, thread priorities, synchronizing threads, inter-thread communication.

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

- Explain errors & exceptions
- Define & create threads, and can implement multithreading, thread priority & synchronization

UNIT-III: (12 Periods)

Collection Framework in Java - Introduction to Java Collections, Overview of Java Collection frame work, Commonly used Collection classes Array List, Vector, Hash table, Stack, Enumeration, Iterator, String Tokenizer, Scanner.
Files - streams - byte streams, character streams, text input/output, binary input/output, random access file operations, File management using File class.
Connecting to Database - JDBC drivers, connecting to a database, querying a database and processing the results, updating data with JDBC.
Learning Outcomes:
At the end of this Unit the student will be able to

- Illustrate appropriate use of collections in solving real world problems.
- Explain basics of streams, stream classes, creation, reading and writing files in context to file handling
- Create Database connection using drivers and can manipulate databases using queries.

UNIT-IV: (10 Periods)

GUI Programming with Java - The AWT class hierarchy, Introduction to Swing, Swing vs AWT, Hierarchy for Swing components, Containers - JFrame, JApplet, JDialog, JPanel, Overview of some swing components Jbutton, JLabel, JTextField, JTextArea, simple swing applications, Layout management - Layout manager types - border, grid and flow layouts.

Learning Outcomes:

At the end of this Unit the student will be able to

- Create GUI applications using AWT and Swing Components.
- Differentiate the AWT and Swing components.

UNIT – V: (8 Periods)

Event handling - Events, Event sources, Event classes, Event Listeners, Relationship between Event sources and Listeners, Delegation event model, Examples: handling a button click, handling mouse events, Adapter classes.

Applets - Inheritance hierarchy for applets, differences between applets and applications, life cycle of an applet, passing parameters to applets, applet security issues.

Learning Outcomes:

At the end of this Unit the student will be able to

- Create GUI applications using Applets.
- Develop GUI applications using event handlers.

TEXT BOOKS:

1. Java Fundamentals - A comprehensive Introduction, Herbet Schidt and Dale Srien, TMH.

REFERENCES BOOKS:

1. Java for Programmers, P.J. Deitel and H.M. Deitel, Pearson education
2. Object Orientd Programming through Java, P. Radha Krishna, Universities Press

ONLINE REFERENCES:

1. https://www.amazon.in/Design-Patterns-Object-Oriented-Addison-Wesley-Professional-ebook/dp/B000SEIBB8

***As Suggested by BOS Experts we are using Java SE 8 for Lab.**

OBJECT ORIENTED PROGRAMMING USING JAVA LAB

COURSE CODEIT226	**CREDITS 1.5**
L T P E O	Sessional Marks: 50
0 0 3 0 3	End Exam Marks: 50
	End Exam:3 Hours

Prerequisite:

Basic knowledge on introduction to programming & object oriented programming concepts are essential.

Course Objectives:

- To understand object oriented programming concepts, and apply them in problem solving.
- To familiarize the concepts of Exception handling, File I/O and Database connectivity.
- To introduce the design of Graphical User Interface using applets and swing controls.

Course Outcomes: After completion of this course student will be able to:

CO-1: Implement OOP'S concepts using Java programming in problem solving
CO-2: Solve problems using Java Files, collections and multithreading
CO-3: Develop simple and complex UI applications using GUI components and databases.

Mapping of course outcomes with program outcomes:

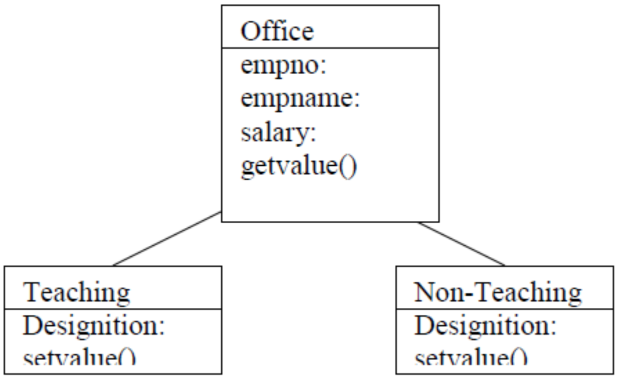
COs/POs-PSOs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2
CO1	3	2	2	2	3				3	1	2	3	2	2
CO2	3	2	2	2	3				3	1	2	3	2	2
CO3	3	1			3							1	3	2

List of Programs:

1. Write a java program to calculate gross salary & net salary taking the following data. [CO-1]

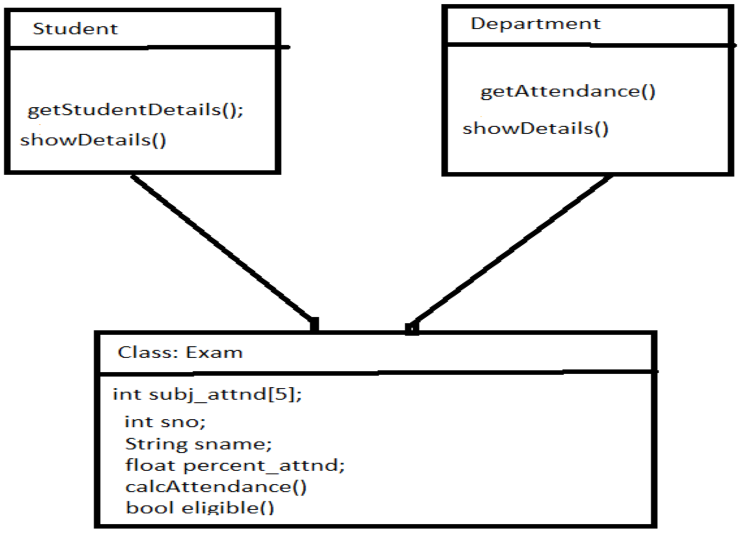
Input :	empno,empname,basic
Process:	DA=50%of basic
	HRA=25%of basic
	PF=10%of basic
	PT=Rs100/-

2. Write a java program that implements educational hierarchy using inheritance. [CO1]



3. Write a program to identify the accessibility of a variable by means of different access specifies within and outside package. [CO1]

4. Write a java program to find the details of the students eligible to enroll for the examination (Students, Department combined give the eligibility criteria for the enrollment class) using interfaces. [CO1]



5. Write a Java program that displays area of different Figures (Rectangle, Square, Triangle) using the method overloading. [CO1]
6. Write a Java program that displays that displays the time in different formats in the form of HH,MM,SS using constructor Overloading. [CO1]
7. Write a Java program that counts the number of objects created by using static variable. [CO1]
8. Write a Java program to count the frequency of words, characters in the given line of text. [CO2]

9. Write a Java program for sorting a given list of names in ascending order. [CO2]
10. Write a Java program that reads a line of integers separated by commas and then displays each integer and fund the sum of he integers (using String Tokenizer). [CO2]
11. Write a Java program that reads a file name from the user then displays information about whether that file exists, file is writable, the type of file and length of the file in bytes. [CO2]
12. Write a Java program that reads a file and displays the file on the screen with a line number before each line. [CO2]
13. Write a Java program that reads a file and displays the no of lines and words in that file. [CO2]
14. Write a Java program that reads to copy source File and display on the console.[CO2]
15. Write a java program that implements Array Index out of bound Exception using built-in-Exception. [CO2]
16. Write a java program that implements bank transactions using user denied exception. [CO2]
17. Write a java program to identify the significance of finally block in handling exceptions. [CO2]
18. Write a java program to generate multiple threads of creating clock pulses.(using runnable interface). [CO2]
19. Write a java program to identify the use of synchronized blocks to synchronized methods. [CO2]
20. Write an applet to display a simple message on a colored background. [CO3]
21. Write an applet to display a moving banner showing the status of it. [CO3]
22. Write an applet to draw a simple and beautiful landscape. [CO3]
23. Write a java program to demonstrate key events by using Delegation event model. [CO3]
24. Write a java program to implement mouse events like mouse pressed, mouse released and mouse moved by means of adapter classes. [CO3]
25. Write a java program to demonstrate window events on frame. [CO3]
26. Write an applet that computes the payment of a loan based on the amount of the loan, interest rate and the number of months. [CO3]
27. Write an applet to perform the 4 basic arithmetic operations as buttons in a form accepting two integers in textboxes and display their result. [CO3]
28. Write a java program to design a registration form for creating a new eMail account. [CO3]
29. Write a java program to design the page authenticating user name and password by using SWING. [CO3]
30. Write a java program to design a calculator by using Grid Layout. [CO3]

OPERATING SYSTEMS LAB

COURSE CODEIT211 **CREDITS 3**

L T P E O
2 1 0 1 4

Sessional Marks: 40
End Exam Marks: 60
End Exam:3 Hours

Prerequisite:

Operating System Concepts.

Course Objectives:

□ Analyze the working of an operating system, its programming interface and filesystem.
□ Develop algorithms for process scheduling, memory management, pagereplacement algorithms and disk scheduling

Course Outcomes:

After completion of this course, a student will be able to :

1. Simulate operating system algorithms for CPU Scheduling and handling deadlocks
2. Implement operating system services like memory management
3. Experiment with Unix-like operating system to interact with kernel through shell scripts.

Mapping of course outcomes with program outcomes:

COs/POs-PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	P010	PO11	PO12	PSO1	PSO2
CO1	3	1	2	2	2				2			1	3	2
CO2	2	1	2	2	1				2			3	3	3
CO3			2	3	2				1			2	3	2

List of Experiments:

1. Shell Programming: a) Unix Commands b) Vi Commands c) Unix Shell programming commands a) Concatenation of two strings b) Comparison of two strings c) Maximum of three numbers d) Fibonacci series e) Arithmetic operation using case (CO3)
2. System Calls a) Process Creation b) Executing a command c) Sleep command d)Sleep command using getpid e) Signal handling using kill k) Wait command i)top (CO3)
3. I/O System Calls a) Reading from a file b) Writing into a file c) File Creation (CO3)
4. a) Implementation of is command b)Implementation of grep command (CO3)
5. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for FCFS and SJF.Print avg.waiting time and turnaround time. (CO1)
6. Given the list of processes, their CPU burst times and arrival times, display/print theGantt chart for Priority and Round robin. Print avg.waiting time and turnaround time. (CO1)

7. a) Implement Bankers algorithm for Dead Lock Avoidance b) Implement an Algorithm for Dead Lock Detection(CO1)
8. Developing Application using Inter Process communication (using shared memory, pipes or message queues) (C01)
9. Producer-Consumer Problem using Semaphore(CO1)
10. Memory management Scheme-I a) Paging Concept(CO2)
11. Memory management Scheme-II a) Segmentation Concept(CO2)
12. Implement the all page replacement algorithms a) FIFO b) LRU c) LFU(CO2)
13. Implement any file allocation technique (Linked, Indexed or Contiguous)(CO3)
14. Linux system administration a. Becoming super user b. Temporarily changing user identity with su command c. Using graphical administrative tools d. Administrative commands e. Administrative configuration files(CO4)
15. Setting up Network File System.(CO3)
16. Firewall and Security Configuration(CO3)

BOOKS:

1. Sumitabha Das, UNIX AND SHELL PROGRAMMING, Tata Mcgraw Hill Publishing Co Ltd
2. W.Richard Stevens, Stephen A.Rago , Advanced programming in the UNIX environment", 3rd Edition Pearson education.
3. Silberschatz, Galvin, and Gagne, “Operating System Concepts”, Sixth Edition, Wiley India Pvt Ltd, 2003.

DATABASE MANAGEMENT SYSTEMS LAB

COURSE CODEIT211 **CREDITS 3**

L T P E O Sessional Marks: 40

2 1 0 1 4 End Exam Marks: 60

End Exam:3 Hours

Prerequisite(s): Fundamentals of computers, familiarity of any one program language

Course Objectives

1. To design and build a simple database system and demonstrate competence withthe fundamental tasks involved with modelling, designing and implementing aDBMS.
2. Perform Pl/SQL programming using concept of Cursor Management, ErrorHandling, Package and Triggers.
3. Understand query processing and techniques involved in queryoptimization.

Course Outcomes

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

CO-1: Design and implement a database schema for a given problem-domain. Query adatabase using SQL DML/DDLcommands.

CO-2: Declare and enforce integrity constraints on a database using RDBMS and optimizethe database using normalizationconcept.

CO-3: Programming PL/SQL including stored procedures, stored functions, cursors,packages.

Mapping of Course Outcomes with POs and PSOs

COs/POs-PSOs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	P010	PO11	PO12	PSO1	PSO2
CO1	3	2	1	1	3				2	2		2	2	1
CO2	3	2	1	1	3				2	2		2	2	1
CO3	3	2	1	1	3				2	2		2	2	1

S.No	Program	Couse Outcome
1	DDL,DML,TCL,DCLcommand	CO1
2	Creating users-roles andGrantingprivileges	CO1
3	Built in functions in oracle (String-Date-Aggregatefunctionsetc)	CO1
4	Implement Integrity Constraints. (Key constraints-Domainconstraints)	CO2
5	Implementing joins-sub queries-nested and correlated nestedqueries	CO2
6	Working with set comparison operators&views	CO2

7	Working with aggregate functions, GROUP BY, HAVINGclauses	CO1
8	Implementing operations usingPL/SQLblocks	CO3
9	Databaseconnectivity	CO3
10	Exceptionhandling	CO3
11	Implementingcursors	CO3
12	Implementingtriggers	CO3
13	Implementing functions and stored procedures&functions -----------------------------	CO3
14	Implementingpackages	CO3
15	Implementing the concepts of Rollback-commitandcheckpoints ----------------------	CO1
16	Design ER Model for a given application & Convert ER model toRelational Model	CO1
17	How to normalize data on the givendatabaseapplication	CO2

TEXT BOOKS

1. Raghurama Krishnan, Johannes Gehrke, “Data base Management Systems”, 3rd Edition, TATAMcGrawHill,2008.
2. Silberschatz, Korth, “Data base System Concepts”, 6th Edition, McGraw Hill, 2010.3.C.J.Date,“Introduction to Database Systems”, 7th Edition,Pearson Education,2002