DATA STRUCTURES

IT211  Credits: 4
Instruction: 4 Periods & 1 Tut /week  Sessional Marks: 40
End- Exam: 3 Hours  End-Exam-Marks: 60

Prerequisite: C Programming

Course Objective:
- 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, a student will be able to:

1. Analyze, evaluate and choose appropriate abstract data types and algorithms to solve particular problems
2. Compare and contrast the benefits of dynamic and static data structures implementations
3. Design and implement abstract data types such as linked list, stack, queue and tree by Using C as the programming language using static or dynamic implementations
4. Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>CO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>PO</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
SYLLABUS

Unit-I: Introduction 12 Periods
Introduction to data structures, arrays and structures. Dynamic Memory Management, Abstract Data Type (ADT). Introduction to Time and Space complexity and their tradeoffs.
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.

Unit-II: Stacks and Queues 12 Periods
Queue ADT: Definition, Primitive operations and Representation. Queue ADT implementation using array and linked list. Types of Queue: Circular Queue, Priority Queue, De-queue Operations and implementation using array and linked list. The queues advantages, disadvantages, and applications.

Unit - III: Sorting and Searching 12 Periods
Sorting: General background, selection sort, bubble sort, insertion sort, shell sort, radix sort, quick sort and merge Sort.
Searching: General background, linear search, binary search and Interpolation search. Introduction to Hashing, Hash Function, Hashing techniques, Collision Resolution Methods: Open Addressing, Chaining.

Unit-IV: Trees 12 Periods
Types: Heap, Binary Search Tree, AVL Tree, B-Tree of order m, introduction to Red-Black tree.

Unit-V: Graphs 16 periods

TEXT BOOKS:
1. Ellis Horowitz, Sartaj Sahni, Fundamentals of Data Structure, computer science Press.

REFERENCE BOOKS:
DIGITAL LOGIC DESIGN  
(COMMON FOR CSE & IT)

IT212
Credits: 3
Instruction: 3 Periods & 1 Tut /week
End- Exam : 3 Hours
Sessional Marks: 40
End-Exam-Marks: 60

Prerequisite:
Computer fundamentals.

Course Objective:
➢ To provide knowledge and understanding of Boolean algebra and digital concepts.
➢ To provide the knowledge of analyzing and designing of combinational and sequential logic networks.
➢ HDL in this course provides the ability to synthesize the designs in Verilog HDL or VHDL.

Course Outcomes:
After completion of this course, a student will be able to:
1. Analyze and synthesize logic circuits by applying the knowledge of number systems, codes, Boolean algebra and digital logic circuits to solve typical problems on the same.
2. Minimize the given Switching function in SOP and POS forms using K-Map & Design of different types of combinational logic circuits using various logic gates.
3. Design and analyze synchronous sequential logic circuits including registers & counters using gates & flip-flops.
4. Design combinational logic circuits using different types of PLDs, namely, PROM, PLA and PAL.

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>PO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
SYLLABUS

UNIT-I: Binary Systems, Boolean Algebra and Logic Gates  10 Periods
Digital Systems, Binary Numbers, Number Systems, Base Conversion Methods, Complements, Signed Binary Numbers, Binary Codes, Binary Logic.

UNIT-II: Gate-Level Minimization  4 Periods
The Map Method, Minimal Functions and their properties, Don't-Care Conditions, Tabulation Method, NAND and NOR Implementation, Other Two- Level Implementations, Verilog Hardware Description Language (Verilog HDL).

Combinational Logic Design:  6 Periods
Combinational Circuits, Analysis Procedure, Design Procedure, Design of adders, subtractors, adder-subtractor circuit, BCD adder circuit, applications of adders, Binary Multiplier, Magnitude Comparator, Decoders, Encoders, Multiplexers, Demultiplexers, Verilog HDL For Combinational Circuits.

UNIT-III: Sequential Logic Circuits  5 Periods
Sequential Circuits, Latches, Flip-Flops, Analysis of Clocked Sequential Circuits, Flip-Flop Conversions, Verilog HDL for Sequential Circuits.

Registers and Counters  6 Periods
Registers, Shift Registers, Ripple Counters, Synchronous Counters, Johnson and Ring counters, Verilog HDL for Registers and Counters.

UNIT-IV: Synchronous Sequential Logic  4 Periods
Basic Design Steps, Serial Adder Example, State Reduction & Assignment Problem.

Fundamentals of Asynchronous Sequential Logic  5 Periods
Introduction, Analysis Procedure, Design Procedure, circuits with latches, Races and Hazards.

UNIT-V: Programmable Logic Devices  8 Periods
Programmable Logic Devices : PROM, PLA, PAL, realization of switching functions using PROM, PLA and PAL; comparison of PROM, PLA and PAL, Programming tables of PROM, PLA and PAL, Sequential Programmable Devices.

Text Books:

Reference Books:
DISCRETE MATHEMATICAL STRUCTURES  
(COMMON FOR CSE & IT)

IT213
Instruction: 4 Periods & 1 Tut /week
End- Exam: 3 Hours

Credits: 4
Sessional Marks: 40
End-Exam-Marks: 60

Prerequisite:

Course Objective:
➢ The knowledge of Mathematics is necessary for a better understanding of almost all the 
   Engineering and Science subjects. Here our intention is to make the students acquainted 
   with the concept of basic topics from Mathematics, which they need to pursue their 
   Engineering degree in different disciplines.

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

1. To understand set theory, relations, mathematical logic, mathematical reasoning and to study 
   about the validity of the arguments.
2. Be able to apply basic counting techniques to solve combinatorial problems.
3. To understand Recurrence Relation, Generating functions and solving problems involving 
   recurrence equations.
4. To familiarize the different types of binary relations and related algorithms on transitive 
   closure.
5. To familiarize with the applications of graphs, trees and algorithms on minimal spanning tress.

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>CO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

PO | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 1 | 2 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
SYLLABUS

UNIT-I: MATHEMATICAL LOGIC  15 Periods

UNIT II: ELEMENTARY COMBINATORICS  8 Periods

UNIT III: RECURRENCE RELATIONS  8 Periods
Generating Functions of Sequences-Calculating their Coefficients-Recurrence relations-Solving recurrence relations-Method of characteristic Roots- Non-homogeneous Recurrence relations and their solutions.

UNIT IV: RELATIONS AND DIGRAPHS  9 Periods

UNIT V: GRAPHS  20 Periods
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.

Text Books:

Reference Books:
COMPUTER ORGANIZATION

IT214
Instruction: 4 Periods & 1 Tut/week
End Exam: 3 Hours

Credits: 4
Sessional Marks: 40
End Exam Marks: 60

Prerequisite:
Computer fundamentals.

Course Objectives:

- Identify and describe the functions of all the basic components making up a computer system.
- Present, as clearly and completely as possible, the characteristics of modern-day computer systems, highlighting on the CPU Organization & Operation, Number systems, Operating Systems Memory Systems, Logic Circuits Design and I/O and Interfacing.
- Engage into contrast discussions based on the two CPU design philosophies i.e. the Complex Instruction Set.
- Computers (CISC) and Reduced Instruction Set Computers (RISC) systems

Course Outcomes:

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

1. Understand Register transfer language, computer instructions and solve problems using micro operations
2. Analyze micro program control to implement micro program instructions
3. Understand central processing unit, stack organization and to evaluate stack operations
4. Review peripheral devices, types of memories and analyze how mapping is done between various memories.

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>CO</th>
<th>PO</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
SYLLABUS

UNIT-I: 14 Periods
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 Algorithm, Decimal Arithmetic Unit.

UNIT-II: 12 Periods
Basic Computer Organization:
Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input-Output and Interrupt, Complete Computer Description.

UNIT-III: 12 Periods
Control Design:
Hardwired & Micro Programmed (Control Unit), Control Memory, Address Sequencing, Conditional and Unconditional Branching, Micro program Example.

UNIT-IV: 12 Periods
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.

UNIT-V: 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, Associative Memory, Cache Memory, Virtual Memory.

TEXT BOOKS:

REFERENCE BOOKS:
DATA COMMUNICATIONS

IT215
Credits: 3
Instruction: 3 Periods & 1 Tut /week
End Exam: 3 Hours

Sessional Marks: 40
End Exam Marks: 60

Prerequisite:
Computer fundamentals.

Course Objectives:
- Introduce students to the evolution of computer networks and the concepts data communication
- Introduce students the general principles of network design and compare the different network topologies
- Introduce students to the digital and analogue representations and channels
- Describe the mechanism and techniques of encoding
- Introduce students to the general principles of circuit and packet switching
- Introduce students to the wireless Local Area Networks
- Provide students with in-depth knowledge of data link layer fundamental such as error detection, correction and flow control techniques; multiple access control techniques

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

1. Understand the basic concepts of Data Communications and different models
2. Understand and analyses the characteristics of signals propagated through different transmission media
3. Apply signal encoding techniques, error detection, correction techniques and learn interfacing
4. Distinguish various Multiplexing techniques and learn various modems like ADSL, xDSL.
5. Illustrate various Data link control protocols namely flow control, error control and HDLC

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>CO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

SYLLABUS

UNIT: I

Data Communication overview: A communication model, Data communications, Data Communication networking- Introduction to WAN, LAN, wireless Networks, MAN, an example configuration Data Transmission: Concepts and Terminology-Transmission terminology, Frequency, spectrum and Bandwidth Analog and Digital Data Transmission-
Introduction to Analog and Digital Data, Analog and Digital Signals, Analog and Digital Transmission, Transmission Impairments-Attenuation, Delay Distortion, Noise, channel Capacity-Nyquist Bandwidth, Shannon Capacity Formula, The expression Eb/N0

**Transmission media:** guided transmission media-Twisted pair, coaxial cable, Optical fiber, Wireless transmission – Antennas, terrestrial microwave, satellite microwave, Broadcast Radio, Infrared Wireless Propagation- Ground wave propagation, sky wave propagation, Line- of-sight Propagation, Line-of-sight Transmission- free space loss, Atmospheric Absorption, Multipath, Refraction

**UNIT: II**

**10 periods**

**Signal Encoding Techniques:** Digital Data Digital signals (Nonreturn to Zero(NRZ), multilevel Binary, Biphas, Modulation rate), Digital Data Analog Signals (Amplitude shift keying, frequency shift keying, Phase Shift keying, Quadrature Amplitude Modulation), Analog Data Digital Signals (Pulse code Modulation, Delta Modulation), Analog Data Analog Signals (Amplitude Modulation, Angle Modulation)

**UNIT: III**

**6 periods**


**UNIT: IV**

**10 periods**

**Multiplexing:** Frequency Division Multiplexing -Characteristics, Analog carrier systems, wave length-Division Multiplexing, Synchronous Time Division Multiplexing-Characteristics, TDM link control, Digital carrier systems, SONET/SDH , Statistical Time Division Multiplexing- characteristics, performance, cable modem, Asymmetric digital subscriber line-ADSL Design, Discrete Multitone, xDSL-HDSL,SDSL,VDSL, modems

**UNIT: V**

**10 periods**

**Data Link Control:** Flow Control-stop and wait flow control, sliding window flow control, Error Control- stop-and-wait ARQ, selective- Reject ARQ , High Level Data Link Control (HDLC) – Basic Characteristics, Frame Structure, operation , Architecture of computer network, layered approach,X.25, Frame relay, ATM.

**Basic hardware:** RJ-45, Network interface card, rack, cable standard-Category 5,6, and 7, cross connection, straight connection cable coding standards.

**Text Books:**

**Reference Books:**
2. Gupta Prakash C.,”Data communication”, PHI Learning
DATA STRUCTURES LAB

IT216
Practical: 3 Periods /week
End Exam: 3 Hours

Credits: 2
Sessional Marks: 50
End Exam Marks: 50

Prerequisite:
C Programming, Data Structures.

Course Objective:
➢ 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, a student will be able to:

1. Implement linear data structures such as stacks, queues, linked lists and apply on real time problem like conversions & evaluations of expressions.
2. Implement non linear data structures such as Trees and Graphs and apply on real time problem like finding shortest path.
3. Implement different sorting and searching techniques.

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>CO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

List of Programs:

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

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

3) Program to do the following
   a) Infix to postfix conversion.  
   b) Evaluation of postfix expression.
4) Programs to implement the following data structures.
   a) Circular Queue       b) Priority Queue

5) Implement primitive operations of de-queue (double ended queue) using a doubly linked list and an array.

6) Program to perform the following operations:
   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.

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

8) Program to implement bfs and dfs for a given graph.

9) ) Program to implement the following sorting methods:
   a) Merge sort            b) Quick sort            c) Insertion Sort     d) Selection Sort

10) ) Program to implement the following searching methods:
   a) Linear Search        b) Binary search

11) 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 hash table.

12) Write a C program to handle the collisions using the following collision resolution Technique
   a) Linear probing       b) Quadratic probing  c) Separate Chaining

Reference Books:
DIGITAL ELECTRONICS LAB
(COMMON FOR CSE & IT)

IT217
Practical: 3 Periods /week
End Exam: 3 Hours

Credits: 2
Sessional Marks: 50
End Exam Marks: 50

Prerequisite:
Digital electronics concepts.

Course Objectives:
➢ To understand how to design and analyze the electronic circuits using semiconductor diodes and operational amplifiers
➢ To understand how to design various combinational and sequential circuits.
➢ To develop and test VHDL Program code for combinational and sequential circuits.

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

<table>
<thead>
<tr>
<th>CO</th>
<th>PO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mapping of course outcomes with program outcomes:
LIST OF LABORATORY EXPERIMENTS

**CYCLE I:**
1. Study of passive, active components & Integrated Circuits.
2. To study the regulation characteristics of given Integrated Circuits.
3. To verify the adder operation & subtractor operation using Operational amplifiers.
4. To verify the truth tables of given Logic Gates.

**CYCLE II:**
1. Verification of truth tables of Logic gates using IC’s.
2. Design a combinational circuit for Code Converters using IC’s.
3. Design a combinational circuit for Adders & Subtractors (HA & FA) using IC’s.
4. Design a sequential circuit for Flip-flop and verify its characteristics using IC’s.
5. Design a bidirectional Universal Shift Register Using IC74LS194.
6. Design of Counters using IC74LS73.

**CYCLE III: (Simulation using VHDL)**
1. Write a program for verification of Basic Gates.
2. Write a program for Adder & Subtractor.
3. Write a program for flip flops.
4. Write a program for MUX & DEMUX.
5. Write a Program for Shift Registers.

*NOTE: FOUR Experiments from each cycle should be done compulsorily.*

Reference Books:
PYTHON PROGRAMMING LAB

IT218

Practical: 3 Periods & 1 Tut /week
End Exam: 3 Hours

Credits: 3
Sessional Marks: 50
End Exam Marks: 50

Prerequisite:
Fundamentals of computers, knowledge in any program language

Course Objective:
- Describe the basic elements of the Python language and the Python interpreter
- Analyze and demonstrate the use of lists, tuples and dictionaries in Python.
- Write classes to demonstrate the ideas of encapsulation, inheritance, interfaces and object oriented program design.
- Explain and demonstrate methods of error handling and Python exceptions.
- Write to and read from files using intermediate file I/O operations in a Python program.
- Solve problems that have origins in a variety of disciplines including math, science, the Internet and business.

Course Outcomes:

<table>
<thead>
<tr>
<th>After completion of this course, a student will be able to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Understand and use the syntaxes of python in problem solving</td>
</tr>
<tr>
<td>2. Apply python data structures to solve real world problems</td>
</tr>
<tr>
<td>3. Implement object oriented concepts in python programming</td>
</tr>
<tr>
<td>4. Demonstrate File I/O and exception handling</td>
</tr>
</tbody>
</table>

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>CO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>
List of the experiments to be done on the following topics

1. **Introduction**: Introduction to Python programming language, using the interpreter, running scripts, variables, assignments, comments, operators and expressions. Introduction to basic data types including strings, integers, lists and tuples.

2. **Control Flow: Conditional** expressions, if statement, for statement and while statement, break and continue statements.

3. **Functions, Methods and Modules**: Introduction to built-in functions, methods and modules. Introduction to standard library modules like sys, os, time and random. Reading command line arguments. Introduction to writing user defined functions and organizing code into modules.

4. **Data structures**: Detailed overview of four major data structures of Python including list, tuple, set and dictionary, including list slicing, sorting lists, list comprehensions.

5. **Working with Files**: Introduction to reading and writing files, text and binary mode. Writing parsers for simple text formats.

6. **Classes and Exceptions**: Introductions to classes, object creation and class inheritance and overriding methods. Introduction to exception handling.

7. **Advanced Topics**

   Introduction to some advanced topics in Python.
   - Downloading things from web
   - Web programming
   - Data visualization
   - Building simple games using pygame

**LIST OF EXPERIMENTS**

**LEVEL 1: FUNDAMENTAL PROGRAMMING**

1. Installation of Python using python interpreter and printing HELLO WORLD message

2. Program that accept the user's first and last name and print them in reverse order with a space between them

3. To display the first and last colors from the following list.
   ```python
color_list = ['Red', 'Green', 'White', 'Black']
   ```

4. To count the number of characters (character frequency) in a string. Sample String: google.com Expected Result : {'o': 3, 'g': 2, '.': 1, 'e': 1, 'l': 1, 'm': 1, 'c': 1}
5. To convert temperatures to and from celsius, fahrenheit.
   [ Formula: $c/5 = f$ 32/9 [ where $c =$ temperature in celsius and $f =$ temperature in fahrenheit ] ] Expected Output: 60°C is 140 in Fahrenheit 45°F is 7 in Celsius

6. To get a list, sorted in increasing order by the last element in each tuple from a given list of non empty tuples.
   Sample List : [(2, 5), (1, 2), (4, 4), (2, 3), (2, 1)]
   Expected Result : [(2, 1), (1, 2), (2, 3), (4, 4), (2, 5)]

7. Write a function translate() that will translate text into “rovarspraket” (Swedish secret language) That is double every consonant and place an occurrence of “o” in between. For example translate(“this is fun”) should return the string “tothohisos isos fofoynon”

8. Program that prints each item and its corresponding type from the following list.
   Sample List : datalist = [1452, 11.23, 1+2j, True, ‘w3resource’, (0, 1), [5, 12], {"class":’V’, "section”:’A’}]}

9. Write a Python function that takes a list and returns a new list with unique elements of the first list. Sample List : [1,2,3,3,3,3,4,5] Unique List : [1, 2, 3, 4, 5]

10. Programs for the following:
    a. Defining and Accessing a Dictionary:
    b. Updating Dictionary:
    c. Deleting Dictionary Elements:
    d. Defining and Accessing, updating ,deleting Tuples.

11. To demonstrate the use of built in string method

12. To demonstrate the use of lists

**LEVEL 2 : CLASSES AND I/O**

13. To implement classes concept in python
14. To implement inheritance in a Banking system
15. To implement polymorphism
16. Python Programs on Exception Handling
    a. Write a python program to handle Number format error
    b. Write a python program to handle IOError
17. Write a python program to perform the following file operations.
    a. Create, open & close a file:
    b. write content on to a file
    c. Read content form the file
    d. Random access operation on files using tell & seek functions
    e. other file operations using the Module ‘os’
LEVEL 3 : PROBLEM SOLVING

18. Cryptography:

In cryptography, a Caesar cipher is a very simple encryption techniques in which each letter in the plain text is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 3, A would be replaced by D, B would become E, and so on. The method is named after Julius Caesar, who used it to communicate with his generals. ROT 13 ("rotate by 13 places") is a widely used example of a Caesar cipher where the shift is 13. In Python, the key for ROT 13 may be represented by means of the following dictionary:

key = {'a':'n', 'b':'o', 'c':'p', 'd':'q', 'e':'r', 'f':'s', 'g':'t', 'h':'u', 'i':'v', 'j':'w', 'k':'x', 'l':'y', 'm':'a', 'n':'b', 'o':'c', 'p':'d', 'q':'e', 'r':'f', 's':'g', 't':'h', 'u':'i', 'v':'j', 'w':'k', 'x':'l', 'y':'m', 'z':'n', 'A':'N', 'B':'O', 'C':'P', 'D':'Q', 'E':'R', 'F':'S', 'G':'T', 'H':'U', 'I':'V', 'J':'W', 'K':'X', 'L':'Y', 'M':'Z', 'N':'A', 'O':'B', 'P':'C', 'Q':'D', 'R':'E', 'S':'F', 'T':'G', 'U':'H', 'V':'I', 'W':'J', 'X':'K', 'Y':'L', 'Z':'M'}

eyour task in this exercise is to implement an encoder/decoder of ROT 13. Once you're done, you will be able to read the following secret message:
Pnrfne pvcure? V zhpu cersre Pnrfne fnynq!
Note that since English has 26 characters, your ROT 13 program will be able to both encode and decode texts written in English.

19. Speech synthesis:

The International Civil Aviation Organization (ICAO) alphabet assigns code words to the letters of the English alphabet acrophonically (Alfa for A, Bravo for B, etc.) so that critical combinations of letters (and numbers) can be pronounced and understood by those who transmit and receive voice messages by radio or telephone regardless of their native language, especially when the safety of navigation or persons is essential. Here is a Python dictionary covering one version of the ICAO alphabet:

d = {'a':'alfa', 'b':'bravo', 'c':'charlie', 'd':'delta', 'e':'echo', 'f':'foxtrot', 'g':'golf', 'h':'hotel', 'i':'india', 'j':'juliet', 'k':'kilo', 'l':'lima', 'm':'mike', 'n':'november', 'o':'oscar', 'p':'papa', 'q':'quebec', 'r':'romeo', 's':'sierra', 't':'tango', 'u':'uniform', 'v':'victor', 'w':'whiskey', 'x':'x ray', 'y':'yankee', 'z':'zulu'}

Your task in this exercise is to write a procedure speak_ICAO() able to translate any text (i.e. any string) into spoken ICAO words. You need to import at least two libraries: os and time. On a mac, you have access to the system TTS (Text To Speech) as follows: os.system('say ' + msg), where msg is the string to be spoken. (Under UNIX/Linux and Windows, something similar might exist.) Apart from the text to be spoken, your procedure also needs to accept two additional parameters: a float indicating the length of the pause between each spoken ICAO word, and a float indicating the length of the pause between each word spoken.
20. Cows and bulls game:

Create a program that will play the “cows and bulls” game with the user. The game works like this:
Randomly generate a 4 digit number. Ask the user to guess a 4 digit number. For every digit that the user guessed correctly in the correct place, they have a “cow”. For every digit the user guessed correctly in the wrong place is a “bull.” Every time the user makes a guess, tell them how many “cows” and “bulls” they have. Once the user guesses the correct number, the game is over. Keep track of the number of guesses the user makes throughout the game and tell the user at the end.
Say the number generated by the computer is 1038. An example interaction could look like this:
Welcome to the Cows and Bulls Game!
Enter a number:
>>> 1234
2 cows, 0 bull
>>> 1256
1 cow, 0 bull
...

21. Chip defect

k defects are randomly distributed amongst n integrated circuit chips produced by a factory (any number of defects may be found on a chip and each defect is independent of the other defects). Let p(k,n) represent the probability that there is a chip with at least 3 defects. For instance p(3,7) ≈ 0.0204081633.

Find p(20 000, 1 000 000) and give your answer rounded to 10 decimal places in the form 0.abcdefghij

Reference Books:
2. David Beazley, Python Cookbook, 3rd edition, O'Reilly Media
   http://chimera.labs.oreilly.com/books/1230000000393/
3. Mark Pilgrim, Dive Into Python 3,
   http://www.diveinto.org/python3/

Other References:
1. Project Euler, https://projecteuler.net/
COMPUTER NETWORKS

IT221
Credits: 3
Instruction: 3 Periods & 1 Tut /week
End Exam: 3 Hours
Sessional Marks: 40
End Exam Marks: 60

Prerequisite:
Data Communications.

Course Objective:
- To provide the students with a sound theoretical and practical knowledge in computer networks.
- To analyze problems associated while connecting components for sharing information.
- To select a protocol stack for specific network.
- To select proper algorithm for the protocols.
- To identify the parameters for real time applications in networks.
- Prepare students for easy transfer from academia to real world.

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

1. Be able to analyze different network architecture’s and designs
2. Mathematically model various error control and routing schemes.
3. Ability to analyze the working of LAN in an organization.
4. Ability to design network architecture for an organization.
5. Ability to design and implement a network for scalability and robustness and security.

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>CO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>PO</th>
<th>PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
SYLLABUS

UNIT – I 10 periods

Protocol Architecture: The need for a protocol architecture, A simple protocol architecture A three layer model, standardized protocol architectures, OSI The model, standardization within the OSI framework, service primitives and parameters, the OSI layers, The TCP/IP protocol Architecture The TCP/IP layers, TCP and UDP Operation of TCP and IP, TCP/IP applications, protocol Interfaces, Local area networks: LAN overview: Background, LAN protocol Architecture LAN standards, IEEE 802, LLC,MAC. Bridges functions, protocol architecture, Fixed routing, spanning tree approach. Layer 2 and Layer 3 switches hubs, layer2 and 3 switches. High speed LANs: The Emergence of High speed LANs, Ethernet MAC, Ethernet, fast Ethernet, gigabit, 10 gbps TokenRing Operation, MAC. Wireless LANs: overview, Wireless LAN Technology, IEEE802.11 Architecture and services, MAC, Physical Layer

UNIT – II 10 periods


UNIT – III 10 periods


UNIT – IV 10 periods

The Transport Layer: The Transport Protocols: connection Oriented transport protocol mechanisms Reliable sequencing network service, unreliable network service, TCP TCP services, TCP header format, TCP Mechanisms, TCP Implementation policy options, TCP congestion control Retransmission timer management, window management, UDP
Unit – V  

**Application Layer:** **Distributed Applications:** Electronic Mail SMTP and MIME  Simple mail transfer protocol (SMTP), multipurpose internet mail extensions (MIME). Hypertext transfer protocol (HTTP)  HTTP overview, messages, request message, response messages, Entities  Network management SNMP  network management systems, SNMPv1, SNMPv2

**Text Books:**

**Reference Books:**
1. Forouzan, “Data communication”, TATA McGraw  
INFORMATION SYSTEMS DESIGN

IT222
Instruction: 3 Periods & 1 Tut /week
End Exam: 3 Hours

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

Prerequisite:
Object Oriented Concepts, C++ programming.

Course Objective:
- On performing a background work prior to begin Project development.
- To gather information and analyze user requirements in system development
- To apply the Process models in developing a project.
- To translate end user requirements into system and software requirements

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

1. Understand the Information systems and systems design.
2. Apply the knowledge of information gathering, Requirement analysis in Software Engineering.
3. Will be able to identify specific components of a software design and use in Interface Designing
4. Use the knowledge of testing and estimate the software development cost

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>CO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Mapping Table:
- CO1 maps with PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PO9, PO10, PO11, PO12
- CO2 maps with PO1, PO2, PO3, PO4, PO5, PO6, PO7, PO8, PO9, PO10, PO11, PO12
- CO3 maps with PO3, PO4, PO5, PO6, PO7, PO8, PO9, PO10, PO11, PO12
- CO4 maps with PO3, PO4, PO5, PO6, PO7, PO8, PO9, PO10, PO11, PO12
SYLLABUS

UNIT I: Text Book 1
Information and Management: Types of Information, Computer based information systems, Management Structure, Management and Information Requirements, Qualities of information (Page No 1 12)
Information Gathering: Strategy to Gather information, Information Sources, Methods of Searching for Information, Interviewing Techniques, Questionnaires Other methods Case Study – Hostel Information System (Page No 34 45)

UNIT – II: Text Book 2
Introduction to Software Engineering: The evolving role of software, Changing Nature of Software, Software myths. (Page No 33 47)
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. (Page No 52 73)
Process models: The waterfall model, Incremental process models, Evolutionary process models, The Unified process. (Page No 77 99)

UNIT – III: Text Book 2
Requirements Engineering: Requirements Engineering Tasks, Initiating the requirements engineering process, Eliciting requirements, developing use cases, Building the analysis model, Negotiating requirements, validating requirements. (Page No 176 204)
Building analysis model: Requirement analysis, Analysis modeling approaches (Page No 208 212)

UNIT – IV: Text Book 2
Design Engineering: Design process and Design quality, Design concepts, the design model. (Page No 261 280)
Performing User interface design: Golden rules, User interface analysis and design, interface analysis, interface design steps, Design evaluation. (Page No 357 382)

UNIT – V: Text Book 2
Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Validation testing, System testing (Page No 387 404, 406 410)
Black Box and White Box testing, Basic Path Testing, Control Structure Testing (Page No 423 434)
Product metrics: Software Quality, A frame work for Product Metrics (Page No 462 471)

Text Books:

Reference Books:
OPERATING SYSTEMS
(COMMON FOR CSE & IT)

IT223
Instruction: 4 Periods & 1 Tut /week
End Exam: 3 Hours

Credits: 4
Sessional Marks: 40
End Exam Marks: 60

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

1. Learn basics of operating system and their structures.
2. Analyze various issues related to inter process communication like process scheduling, resource management and deadlocks.
3. Interpret the issues and challenges of memory management.
4. Understand the concepts of I/O management, file system implementation and problems related to security and protection.

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>CO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SYLLABUS

UNIT – I
INTRODUCTION TO OS
Introduction to operating systems – operating system structures – system calls – system structure – virtual machines.
PROCESS MANAGEMENT

UNIT – II
PROCESS SCHEDULING AND SYNCHRONIZATION

UNIT – III
MEMORY MANAGEMENT

UNIT – IV
FILE SYSTEMS AND ITS IMPLEMENTATION

UNIT – V
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.

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:

Reference Books:
PROBABILITY STATISTICS & QUEUING THEORY

IT224
Instruction: 4 Periods & 1 Tut /week
End Exam: 3 Hours

Credits: 4
Sessional Marks: 40
End Exam Marks: 60

Prerequisite:
Engineering Mathematics 1, 2 & Discrete Mathematical Structures.

Course Objective:
➢ The knowledge of Mathematics is necessary for a better understanding of almost all the Engineering and Science subjects. Here our intention is to make the students acquainted with the concept of basic topics from Mathematics, which they need to pursue their Engineering degree in different disciplines.

Course Outcomes:

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

1. Understand the concepts of various statistical measures like mean, variance and standard deviation of a random variable.
2. Familiarize the different types probability distributions and their properties.
3. Compute simple correlation between the variables and fit straight line, parabola by the principle of least squares.
4. Analyze the statistical data and apply various small or large sample tests for testing the hypothesis.
5. Learn about different Queuing models and its applications.

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>CO</th>
<th>PO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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. Random Variables: Discrete and Continuous random variables, Distribution function of random variable, Properties, Probability mass function, Probability density function, Mathematical expectation, Properties of Mathematical expectations, Mean and Variance.

UNIT – II: PROBABILITY DISTRIBUTION 14 Periods

UNIT – III: CURVE FITTING , CORRELATION AND REGRESSION 10 Periods
Curve Fitting: Principle of Least Squares, Method of Least Squares (Straight Line and Parabola).
Correlation: Definition, Measures of correlation, Correlation for Bivariate Distribution, Rank correlation coefficients.
Regression: Simple linear regression, regression lines and properties.

UNIT – IV: TESTING OF HYPOTHESIS 14 Periods
Formulation of Null Hypothesis, Critical Region, Level of Significance.
Small Samples: Students t distribution (Significance test of a sample mean, Significance test of difference between sample means), F distribution, $\chi^2$ test, Goodness of fit.
Large samples: Test of Significance of Large Samples – Single Proportion, Difference between two Proportions, Single mean and Difference of means.

UNIT – V: QUEUEING THEORY 10 Periods
Queue description, characteristics of a queuing model, study state solutions of M/M/1: $\alpha$ Model, M/M/1; N Model.

TEXT BOOK:

REFERENCE BOOKS:
COMPUTER GRAPHICS & MULTIMEDIA

IT225
Instruction: 3 Periods & 1 Tut /week
End Exam: 3 Hours

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

Prerequisite:
Computer fundamentals, C programming.

Course Objective:
➢ This course provides an introduction to the principles of computer graphics. In particular, the course will consider methods for modeling 2 dimensional & 3 dimensional objects and efficiently generating photorealistic renderings on color raster graphics devices. The emphasis of the course will be placed on understanding how the various elements that underlie computer graphics (algebra, geometry, algorithms and data structures, optics, and photometry) interact in the design of graphics software systems.

Course Outcomes:

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

1. Understand Computer graphics applications and apply algorithms to obtain output primitives..
2. Apply Geometric Transformations on multimedia.
3. Students will understand the concepts and techniques used in 3D computer graphics.
4. Apply 2D and 3D multimedia building blocks to Develop multimedia applications.

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>CO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
SYLLABUS

UNIT – I: OUTPUT PRIMITIVES 10 hours
Introduction  Line  Circle and Ellipse Drawing Algorithms – Attributes – Two Dimensional Geometric Transformations – Two Dimensional Clipping and Viewing.

UNIT – II: THREE DIMENSIONAL CONCEPTS 8 hours

UNIT III: MULTIMEDIA SYSTEMS DESIGN 10 hours

UNIT – IV: MULTIMEDIA FILE HANDLING 10 hours

UNIT – V: HYPERMEDIA 10 hours

Text Books:

Reference Books:
NETWORKING LAB

IT226
Practical: 3 Periods/Week
End Exam: 3 Hours

Credits: 2
Sessional Marks: 50
End Exam Marks: 50

Prerequisite:
Computer Networks Concepts.

Course Objectives:

➢ The objective of this lab is to introduce students to the design issues that arise in building and using networks and to give students hands on experience with building and using network services.

➢ The practical issues to be stressed include design and installation of LAN, network operating system, setting up a network system such as users and their permissions and rights, groups and domains, adding workstations and sharing of resources across the network

Course Outcomes:

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

1. Understand and identify the various network infrastructure and command needed for network design and troubleshooting.

2. Understand the basic concepts and functions of Layer 1 (Hubs), Layer 2(Switches and bridges) and Layer 3 (Router).

3. Understand the building components of network design.

4. Understand the basic format of known protocols such as TCP, UDP, ICMP..Etc.

5. Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>CO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

LIST OF EXPERIMENTS

I. Study Experiments:

This study experiments helps the learners to understand certain network components like Hubs, switches, routers, wireless access modems, transmission medium (coaxial cables, twisted pair cables, optical fiber) and several networking components

2 weeks duration
1. Study of specifications of latest desktops and laptops
2. Familiarization with Networking Components and devices: LAN Adapters, Hubs, Switches, routers etc.
3. Familiarization with Transmission media and Tools: Coaxial cable, UTP Cable, Crimping tool, Connectors etc.
4. Study of various LAN topologies and their creation using network devices, cables and computers
5. Study of Client Server Architecture
6. To study LAN using bus, tree, star topology
7. To study pc to pc communication using parallel port
8. To study fiber optics communication
9. To study wireless communication

II. Hands on Experiments
8 weeks duration
This set of experiments helps the learners in gaining expertise in developing and maintaining a certain network which includes setting up a LAN network and maintaining it, configuring routers, switches and firewalls using a certain Hardware components.
1. preparing straight and cross cables.
2. Study of network commands and network configuration commands
3. Implementation of file and printer sharing
4. Designing and implementing Class A, B, and C Networks
5. Subnet planning and its implementation.
6. To configure the IP address for a computer connected to LAN and to configure network parameters of a web browser for the same computer.
7. To install any one open source packet capture software like wire shark etc.
8. To configure WLAN
9. To install and configure wireless access points
10. To configure modem of a computer
11. To configure hub/switch and router
13. Configuring a router based firewall

III. Programming Experiments
5 weeks duration
This set of programming experiments helps the learners in simulating different routing protocols, network topologies and several layered protocols using simulators like NS2 and packet tracing software’s
1. Configure a network topology using packet tracing software
2. Configure a network using Distance vector routing protocol using packet tracer software
3. Static routing using packet tracer software
4. DHCP, DNS, HTTP configuration using packet tracer software

Experiments beyond the Syllabus:
1. Developing a VPN network for number of 50 users
2. TCP, UDP protocol simulation using NS2

Reference Books:
1. CCNA Study guide.
COMPUTER GRAPHICS & MULTIMEDIA LAB

IT227
Practical: 3 Periods/Week
End Exam: 3 Hours

Credits: 2
Sessional Marks: 50
End Exam Marks: 50

Prerequisite:

Course Objective:
➢ The computer graphics and multimedia laboratory is established for the purpose of providing
working area for development of computer graphics and multimedia

Course Outcomes:

<table>
<thead>
<tr>
<th>After completion of this course, a student will be able to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Draw various types of lines and curves.</td>
</tr>
<tr>
<td>2. Create animations using various editing tools</td>
</tr>
<tr>
<td>3. Use audio, video, internet editing tools to develop multimedia applications</td>
</tr>
</tbody>
</table>

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>PO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

LIST OF EXPERIMENTS
1. To implement Bresenham’s algorithms for line, circle and ellipse drawing
2. To perform 2D Transformations such as translation, rotation, scaling, Reflection and
sharing.
3. To implement Cohen–Sutherland 2D clipping and window–viewport mapping
4. To perform 3D Transformations such as translation, rotation and scaling.
5. User Interface Design & Graphics II: Create a user interface for your final project. Include 2
backgrounds and 1 button set. Aim for a cohesive look.
6. Multimedia Sound: Create 2 soundtracks and 2 EFX sounds for a previous project.
7. Procedure to create an animation to indicate a ball bouncing on steps
8. Procedure to simulate movement of a cloud.
9. Procedure to create an animation with the following features. WELCOME Letters should
appear one by one the fill color of the text should change to a different color after the
display of the full word.
10. Procedure to create an animation to represent the growing moon
11. Procedure to extract the flower only from given photographic image and organize it on a
background. Selecting your own background for organization.
12. Procedure to use appropriate tool(s) from the toolbox cut the objects from 3 files (f1.jpg, 
f2.jpg & f3.jpg); organize them in a single file and apply feather effects.
Reference Books:

OPERATING SYSTEMS (LINUX) LAB

IT228

Practical: 3 Periods/Week
End Exam: 3 Hours

Credits: 2
Sessional Marks: 50
End Exam Marks: 50

Prerequisite:
Operating System Concepts.

Course Objectives:
- Analyze the working of an operating system, its programming interface and file system.
- Develop algorithms for process scheduling, memory management, page replacement algorithms and disk scheduling

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

1. Implement scheduling algorithms, deadlock management.
2. Implement free space management and page replacement strategies.
3. Implement file allocation methods and disk scheduling algorithms.

Mapping of course outcomes with program outcomes:

<table>
<thead>
<tr>
<th>CO</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>3</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

List of Experiments:
1. Shell Programming & AWK scripts
2. Write programs using the following system calls of LINUX operating system: Fork, exec, getpid, exit, wait, close, stat, opendir, readdir
3. Write programs using the I/O system calls of LINUX operating system (open, read, write, etc) and error reporting using errno
4. Write C programs to simulate UNIX commands like ls, grep, etc.
5. Given the list of processes, their CPU burst times and arrival times, display/print the Gantt chart for scheduling algorithms FCFS, SJF, PRIORITY & RR. For each of the scheduling policies, compute and print the average waiting time, average turnaround time and Gantt chart
7. Programs using pipes
8. Implement Banker’s algorithm for handling deadlock
9. Implement free space management strategies such as First fit, Best fit and Worst fit
10. Implement page replacement algorithms such as FIFO, LRU
11. Implement file allocation techniques (Linked, Indexed and Contiguous)
12. Implement disk arm scheduling algorithms such as FCFS, SSTF
**Reference Books:**
1. Sumitabha Das, **UNIX AND SHELL PROGRAMMING**, Tata Mcgraw Hill Publishing Co Ltd
2. Yashwanth Kanetkar, **UNIX shell programming**, BPB publications