III SEMESTER

Semester: III			
Discrete Mathematical Structures and Probability			
Course Code:	**MVJ21MA31B**		**CIE Marks:50**
Credits:	**L:T:P:S:3:2:0:0**		**SEE Marks: 50**
Hours:	**30L+20T**		**SEE Duration: 3 Hrs**

Course Learning Objectives: The students will be able to	
1	Prepare for a background in abstraction, notation, and critical thinking for the mathematics most directly related to computer science.
2	Understand and apply mathematical induction, combinatorics, discrete probability, sequence and recurrence, elementary number theory.
3	Understand and apply probability distribution, sampling theory and joint probability distributions.

UNIT-I	
Properties of the Integers: Mathematical Induction. **Principles of Counting**: Fundamental Principles of Counting, The Rules of Sum and Product, Permutations, Combinations – The Binomial and Multinomial Theorem, Combinations with Repetition. **Self-Study Topic:** The Well Ordering Principle. **Video Link**: 1. http://nptel.ac.in/courses.php?disciplineID=111	**10 Hrs**
UNIT-II	
The Principle of Inclusion and Exclusion: The Principle of Inclusion and Exclusion, Generalizations of the Principle. Derangements – Nothing is in its Right Place, Rook Polynomials. **Recurrence Relations:** First Order Linear Recurrence Relation, The Second Order Linear Homogeneous Recurrence Relation with Constant Coefficients. **Self-Study Topic:** Non-Homogeneous Recurrence Relation **Video Link**: 1. http://nptel.ac.in/courses.php?disciplineID=111	**10 Hrs**
UNIT-III	
Relations: Cartesian Products, Relations, Properties of Relations, Equivalence Relations Zero-One Matrices and Directed Graphs, Partial Orders – Hasse Diagrams and extreme elements.	**10 Hrs**

Functions: Plain and One to One, Onto Functions. The Pigeon-hole Principle, Function Composition, and Inverse Functions. **Self-Study Topic:** Lattice **Video Link**: 1. http://nptel.ac.in/courses.php?disciplineID=111	
UNIT-IV	
Probability Distributions: Random variables (discrete and continuous), probability mass/density functions. Binomial distribution, Poisson distribution. Exponential and normal distributions, problems. **Joint probability distribution:** Joint Probability distribution for two discrete random variables, expectation, covariance, correlation coefficient. **Self-Study Topic:** Continuous Joint Probability Distribution. **Video Link**: 1. http://nptel.ac.in/courses.php?disciplineID=111	**10 Hrs**
UNIT-V	
Sampling Theory: Sampling, Sampling distributions, standard error, test of hypothesis for means and proportions, confidence limits for means, student's t-distributionand Chi-square distribution. **Coding Theory:**Coding of binary information and error detection. **Self-Study Topic:** Decoding and error detection. **Video Link**: 1. http://nptel.ac.in/courses.php?disciplineID=111	**10 Hrs**

Course Outcomes: After completing the course, the students will be able to	
CO1	Demonstrate the application of discrete structures in different fields of computer Science.
CO2	Solve problems using recurrence relations and generating functions.
CO3	Solving logical problem using concepts of relations and functions.
CO4	Develop probability distribution of discrete, continuous random variables and joint probability distribution occurring in digital signal processing, information theory and Design engineering.
CO5	Demonstrate testing of hypothesis of sampling distributions.

Reference Books	
1.	Ralph P. Grimaldi: Discrete and Combinatorial Mathematics, 5th Edition, Pearson Education.2004.
2.	B.S. Grewal, "Higher Engineering Mathematics" Khanna Publishers, 43rd Edition, 2013.
3.	Ramana B. V., "Higher Engineering Mathematics", Tata Mc Graw-Hill, 2006.
4.	Kenneth H. Rosen: Discrete Mathematics and its Applications, 6thEdition, McGraw Hill, 2007

Continuous Internal Evaluation (CIE):

Theory for 50 Marks

CIE is executed by way of quizzes (Q), tests (T) and assignments. A minimum of three quizzes are conducted along with tests. Test portion is evaluated for 50 marks and quiz is evaluated for 10 marks. Faculty may adopt innovative methods for conducting quizzes effectively. The number of quizzes may be more than three (conduct additional quizzes and take best three). The three tests are conducted for 50 marks each and the average of all the tests are calculated for 50. The marks for the assignments are 20 (2 assignments for 10 marks each). The marks obtained in test, quiz and assignment are added to get marks out of 100 and report CIE for 50 marks.

Semester End Examination (SEE):

Total marks: 50+50=100

SEE for 50 marks is executed by means of an examination. The Question paper for each course contains two parts, Part – A and Part – B. Part – A consists of objective type questions for 20 marks covering the entire syllabus. Part – B Students have to answer five questions, one from each unit for 16 marks adding up to 80 marks. Each main question may have a maximum of three sub divisions. Each unit will have internal choice in which both questions cover entire unit having same complexity in terms of COs and Bloom's taxonomy level.

CO-PO Mapping												
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	0	2	0	0	0	0	0	0	0	1
CO2	2	3	0	2	0	0	0	0	0	0	0	1
CO3	2	3	0	3	0	0	0	0	0	0	0	1
CO4	3	3	0	3	0	0	0	0	0	0	0	1
CO5	3	3	0	3	0	0	0	0	0	0	0	1

High-3, Medium-2, Low-1

Semester: III		
OBJECT ORIENTED PROGRAMMING		
Course Code: MVJ21AI32		**CIE Marks:100**
Credits: L:T:P:S:3:1:0:0		**SEE Marks: 100**
Hours: 40L+26T		**SEE Duration: 3 Hrs**

Course Learning Objectives: The students will be able to	
1	Identify the need for Java - an object oriented language. Set up Java JDK environment to create, debug and run simple Java programs.
2	Illustrate the use of classes and distinguish the usage of different types of Inheritance and constructors in real world.
3	Demonstrate the use of exceptions and to create multi-threaded programs
4	Illustrate the use of Collections with elements in Java program.
5	Develop Java Application using JDBC connectivity.

UNIT-I	
Prerequisites : Basic Knowledge about C or C++ **Introduction to Object Oriented Concepts and Java:** Java's Magic: the Byte code; Java Development Kit (JDK); The Java Buzz words, Object Oriented Programming - Two Paradigms, Abstraction, The Three OOP Principles and its advantages, Simple Java programs. Data types, variables and arrays, Operators, Control Statements. **Laboratory Sessions/ Experimental learning:** A professor in college will allow a student to be excused from the final exam if either of the following is true: • They have a 90% average or higher in the class and have missed 3 or less class lectures. • They have a 80% average or higher in the class and have not missed any class lectures. The program below will determine whether a student can get out of the exam or not. Rewrite the program so only one if statement is used. **Applications:** Arrays in mathematical vectors, matrices. **Video link / Additional online information (related to module if any):** • Differences between JVM vs JRE vs JDK in Java: https://www.youtube.com/watch?v=5Bp6GLU6HKE	**10 Hrs**
UNIT-II	
Classes, Inheritance, Packages and Interfaces: Classes fundamentals; Declaring objects; Assigning object reference variables; Introducing Methods, Constructors, this keyword, Finalize Method. Inheritance: Inheritance basics, using super, creating multi-level hierarchy, when constructors are called, method overriding, using abstract classes.	**10 Hrs**

Packages, Access Protection, Importing Packages, Interfaces. **Laboratory Sessions/ Experimental learning:** Write a program that calculates the number of buckets of paint to use for a room and the optimal number of cans to purchase. You need to ask the height of the room and the length and width of the room. The room is rectangular. You must paint the walls and the ceiling but not the floor. There are no windows or skylights. You can purchase the following size buckets of paint. • 5-liter bucket costs $15 each and covers 1500 square feet. • 1-liter bucket costs $4 and covers 300 square feet. **Applications:** Inheritance in Banking Sectors **Video link / Additional online information (related to module if any):** Types of Inheritance: https://www.youtube.com/watch?v=ZP27c7i5zpg	
UNIT-III	
Exception Handling and Multi-Threaded Programming: Exception Handling fundamentals, Exception Types, Uncaught Exceptions, Using try catch, Multiple catch clauses, Nested try statements, throw, throws, finally, Java's built-in exceptions, Programming Examples. **Multi-Threaded Programming**: The java thread model, Main thread, Creating Thread, Creating multiple threads, Using is Alive() and join(), Thread priorities, Synchronization; InterThread Communication - Bounded buffer problem. **Laboratory Sessions/ Experimental learning:** The Producer-Consumer problem describes two processes, the producer and the consumer, which share a common, fixed-size buffer used as a queue. The producer's job is to generate data, put it into the buffer, and start again.At the same time, the consumer is consuming the data (i.e. removing it from the buffer), one piece at a time. Make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer. Write a java code to get the solution for this multi-process synchronization problem. **Applications:** Multithreads in Browsers, Servers **Video link / Additional online information (related to module if any):** Multithreading: https://www.youtube.com/watch?v=O_Ojfq-OIpM	**10 Hrs**
UNIT-IV	
The collections and Framework: Collections Overview, Recent Changes to Collections, The Collection Interfaces, The Collection Classes, Accessing a collection Via an Iterator, Storing User Defined Classes in Collections. **Java Lambda expressions:** Java Lambda expressions, Using Java Lambda expressions,	**10 Hrs**

Lambda expression vs method in java, Lambda expression in the array list. **Laboratory Sessions/ Experimental learning:** Write a Java program to iterate through all elements in a array list **.** Write a Java program to create a new array list, add some colors (string) and print out the collection **Applications:** Elements in group **Video link / Additional online information (related to module if any):** https://www.youtube.com/watch?v=Q_9vV3H-dt4	
UNIT-V	
JDBC: The Concept of JDBC; JDBC Driver Types; JDBC Packages; A Brief Overview of the JDBC process; Database Connection; Associating the JDBC/ODBC Bridge with the Database; Statement Objects; Result Set; Transaction Processing; Metadata, Data types; Exceptions. **Laboratory Sessions/ Experimental learning:** Develop Student Management System application with swings as the front end and database as the back end using JDBC connectivity. **Applications:** Scientific Applications, Financial Applications **Video link / Additional online information (related to module if any):** **Java JDBC :**https://www.youtube.com/watch?v=hEWBIJxrLBQ	**10 Hrs**

Course Outcomes: After completing the course, the students will be able to	
CO1	Illustrate the Object Oriented Programming concepts and basic characteristics of Java.
CO2	Demonstrate the principles of classes, inheritance, packages and interfaces.
CO3	Experiment with exception handling Mechanisms and Create multi-threaded programs.
CO4	Interpret the need for advanced Java concepts like collections in developing modular and efficient programs.
CO5	Develop an application with Database using JDBC connectivity.

Reference Books	
1.	Herbert Schildt, Java The Complete Reference, 7 /9th Edition, Tata McGraw Hill, 2007.
2.	Jim Keogh: J2EE-The Complete Reference, McGraw Hill, 2007.
3.	Effective Java, Third Edition, Joshua Bloch, Addison-Wesley Professional,2017
4.	Richard Warburton**,** Java 8 Lambdas: Pragmatic Functional Programming Kindle Edition.

Continuous Internal Evaluation (CIE):

Theory for 50 Marks

CIE is executed by way of quizzes (Q), tests (T) and assignments. A minimum of three quizzes are conducted along with tests. Test portion is evaluated for 50 marks and quiz is evaluated for 10 marks. Faculty may adopt innovative methods for conducting quizzes effectively. The number of quizzes may be more than three (conduct additional quizzes and take best three). The three tests are conducted for 50 marks each and the average of all the tests are calculated for 50. The marks for the assignments are 20 (2 assignments for 10 marks each). The marks obtained in test, quiz and assignment are added to get marks out of 100 and report CIE for 50 marks.

Semester End Examination (SEE):

Total marks: 50+50=100

SEE for 50 marks is executed by means of an examination. The Question paper for each course contains two parts, Part – A and Part – B. Part – A consists of objective type questions for 20 marks covering the entire syllabus. Part – B Students have to answer five questions, one from each unit for 16 marks adding up to 80 marks. Each main question may have a maximum of three sub divisions. Each unit will have internal choice in which both questions cover entire unit having same complexity in terms of COs and Bloom's taxonomy level.

CO-PO/PSO Mapping														
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	1	-	-	-	-	-	-	-	-	3	2	-
CO2	3	3	1	-	-	-	-	-	-	-	-	3	2	-
CO3	3	3	1	2	-	-	-	-	-	1	-	3	2	3
CO4	3	3	3	3	-	-	-	2	2	2	-	3	2	-
CO5	3	3	3	3	-	-	2	2	3	2	-	3	2	3

High-3, Medium-2, Low-1

Semester: III		
OPERATING SYSTEMS		
Course Code: MVJ21AI33		**CIE Marks:100**
Credits: L:T:P:S:3:1:0:0		**SEE Marks: 100**
Hours: 40L+26T		**SEE Duration: 3 Hrs**

Course Learning Objectives: The students will be able to	
1	Introduce concepts and terminology used in OS.
2	Explain threading and multithreaded systems.
3	Illustrate process synchronization and concept of Deadlock.
4	Introduce Memory and Virtual memory management, File system and storage techniques.

UNIT-I	
Introduction: What operating systems do; Computer System organization; Computer System architecture; Operating System operations; Distributed system; Special-purpose systems; Computing environments. Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines; System boot. **Process Management**: Process concept; Process scheduling; Operations on processes; Inter process communication.	**8 Hrs**
UNIT-II	
Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling. **Process Synchronization**: Synchronization: The critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.	**8 Hrs**
UNIT-III	
Deadlocks : Deadlocks; System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock. **Memory Management**: Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation	**8 Hrs**

UNIT-IV	
Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing. **File System, Implementation of File System**: File system: File concept; Access methods; Directory structure; File system mounting; File sharing; **Implementing File system**: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.	**8 Hrs**
UNIT-V	
Mass Storage Structure-Disk Structure-Disk Attachment-Disk Scheduling-Disk Management- Swap-Space Management. **Protection**: Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability- Based systems. Case Studies: **Windows, Unix, Linux, Android.**	**8 Hrs**

Course Outcomes: After completing the course, the students will be able to	
CO1	Illustrate the fundamental concepts of operating systems
CO2	Compare and illustrate various process scheduling algorithms.
CO3	Ability to recognize and resolve Deadlock problems, Memory Management techniques.
CO4	Apply appropriate memory and file management schemes.
CO5	Appreciate the need of access control and protection in Operating System and illustrate various disk scheduling algorithms.

Reference Books	
1.	Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Concepts 7th edition,Wiley-India, 2006
2.	D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw- Hill, 2013.
3.	Tanenbaum, A., "Modern Operating Systems", Prentice-Hall of India. 2004
4.	P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition

Continuous Internal Evaluation (CIE):

Theory for 50 Marks

CIE is executed by way of quizzes (Q), tests (T) and assignments. A minimum of three quizzes are conducted along with tests. Test portion is evaluated for 50 marks and quiz is evaluated for 10 marks. Faculty may adopt innovative methods for conducting quizzes effectively. The number of quizzes may be

more than three (conduct additional quizzes and take best three). The three tests are conducted for 50 marks each and the average of all the tests are calculated for 50. The marks for the assignments are 20 (2 assignments for 10 marks each). The marks obtained in test, quiz and assignment are added to get marks out of 100 and report CIE for 50 marks.

Semester End Examination (SEE):

Total marks: 50+50=100

SEE for 50 marks is executed by means of an examination. The Question paper for each course contains two parts, Part – A and Part – B. Part – A consists of objective type questions for 20 marks covering the entire syllabus. Part – B Students have to answer five questions, one from each unit for 16 marks adding up to 80 marks. Each main question may have a maximum of three sub divisions. Each unit will have internal choice in which both questions cover entire unit having same complexity in terms of COs and Bloom's taxonomy level.

CO-PO/PSO Mapping														
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	2	-	-	-	-	-	-	-	-	-	2	-
CO2	2	2	3	-	-	-	-	-	-	-	-	-	2	-
CO3	3	2	3	-	-	-	-	-	-	-	-	-	3	-
CO4	3	2	3	-	-	-	-	-	-	-	-	-	2	2
CO5	3	2	3	-	-	-	-	-	-	-	-	-	2	-

High-3, Medium-2, Low-1

Semester: III		
DATA STRUCTURES AND APPLICATIONS & LAB		
Course Code: MVJ21AI34		**CIE Marks:50+50**
Credits: L:T:P: 3:0:1		**SEE Marks: 50 +50**
Hours:40 L+ 26 P		**SEE Duration: 03+03 Hours**

Course Learning Objectives: The students will be able to	
1	Identify the importance of data structures & memory allocation.
2	Perform operations on stacks and queues and its applications.
3	Apply the operations of linked list, Trees & Graphs in various applications.
4	Apply searching and sorting operations in real time applications.
5	Identify the importance of data structures & memory allocation.

UNIT-I	

Introduction: Data Structures, Classifications (Primitive &Non Primitive), Data structure Operations, Review of Arrays, Structures, Self-Referential Structures. Pointers and Dynamic Memory Allocation Functions. Representation of Linear Arrays in Memory, Dynamically allocated arrays.
Abstract Data Type, Array Operations: Traversing, inserting, deleting, searching, and sorting**,**
Array ADT :Multidimensional Arrays, Polynomials and Sparse Matrices.
Strings: Basic Terminology, Storing, Operations and Pattern Matching algorithms. Programming Examples.
Laboratory Sessions/ Experimental learning:
1. Create an array of structure which has the following members Student name, Student USN, Marks1, Marks2, Marks3. Allocate memory to store 5 students details initially. When a new student details need to be entered or to be deleted in this array, dynamically change the array size. Write a program to implement this scenario and display the result.
2. Find the bug for the following code and then Debug it
int minval(int *A, int n) {
int currmin;
for (int i=0; i<n; i++)
if (A[i] <currmin)
	10 Hrs

```
currmin = A[i];
        return currmin;
      }
```

3. Compile the following code and debug it.

```
#include <stdio.h>
#include <string.h>
struct student
{
 int id;
 char name[30];
 float percentage;
};
int main()
{
 int i;
 struct student record1 = {1, "Raju", 90.5};
 struct student *ptr;
printf("Records of STUDENT1: \n");
printf("  Id is: %d \n", ptr->id);
printf("  Name is: %s \n", ptr->name);
printf("  Percentage is: %f \n\n", ptr->percentage);
 return 0;
}
```

Real Time Applications: System memory allocation

Video link / Additional online information (related to module if any):

1. https://nptel.ac.in/courses/106106130/ 2. https://nptel.ac.in/courses/106105085/ 3. https://nptel.ac.in/courses/106/106/106106127/ 4. https://www.coursera.org/lecture/data-structures/arrays-OsBSF	
UNIT-II	
Stacks: Definition, Stack Operations, Stack ADT, Array Representation of Stacks, Stacks using Dynamic Arrays, Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression. **Recursion -** GCD, Tower of Hanoi. **Queues:** Definition, Array Representation, Queue Operations, Queue ADT, Circular Queues, Circular queues using Dynamic arrays, Dequeues, Priority Queues. Programming Examples. **Laboratory Sessions/ Experimental learning:** Design, Develop and Implement a menu driven Program in C for the following operations on DEQUEUE of Integers (Array Implementation of Queue with maximum size MAX) a. Insert an Element on to DEQUEUE b. Delete an Element from DEQUEUE c. Demonstrate Overflow and Underflow situations on DEQUEUE d. Display the status of DEQUEUE e. Exit Support the program with appropriate functions for each of the above operations **Real Time Applications: Game applications, Ticket booking applications (Eg: Train, restaurant etc)** **Video link / Additional online information (related to module if any):** 1.https://nptel.ac.in/courses/106106130/ 2. https://nptel.ac.in/courses/106102064/ 3. https://nptel.ac.in/courses/106105085/ 4.https://nptel.ac.in/courses/106/106/106106127/	**10 Hrs**
UNIT-III	
Linked Lists: Definition, Representation of linked lists in Memory, Memory allocation; Garbage Collection. Linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists, Circular linked lists, and header linked lists. Linked Stacks and Queues. Applications of Linked lists – Polynomials. Programming Examples **Hashing:** Hash Table organizations, Hashing Functions, Static and Dynamic Hashing. **Laboratory Sessions/ Experimental learning:** 1.Design, Develop and Implement a Program in C for the following operations on Singly	**10 Hrs**

Circular Linked List (SCLL) with header nodes
a. Represent and Evaluate a Polynomial $P(x,y,z) = 6x^2 y^2 z-4yz^5 +3x^3 yz+2xy^5 z-2xyz^3$
b. Find the sum of two polynomials POLY1(x,y,z) and POLY2(x,y,z) and store the result in POLYSUM(x,y,z) Support the program with appropriate functions for each of the above operations
2. Debug the following code and explain the process
 	
//Insert a value into an ordered linked list
 	
void insert(lnode*&curr, int val) {
 	
if (curr == NULL)
 	
curr = new lnode(val, NULL);
 	
else if (lnode->val>val)
 	
curr = new lnode(val, curr->next);
 	
else {
 	
curr = curr->next;
 	
insert(curr, val);
 	
}
 	
}
 	
Real Time Applications: Music Player, Image Viewer, Web browser, Process Management, Mechanical field
 	
Video link / Additional online information (related to module if any):
 	
1. https://nptel.ac.in/courses/106106130/
2. https://nptel.ac.in/courses/106102064/
3. https://nptel.ac.in/courses/106105085/	

UNIT-IV	
Trees: Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals - Inorder, postorder, preorder; Additional Binary tree operations. Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching, Application of Trees-Evaluation of	**10 Hrs**

Expression, AVL Trees, Splay Trees, B-Tree, Programming Examples **Laboratory Sessions/ Experimental learning:** Design, Develop and Implement a menu driven Program in C for the following operations on AVL Trees i) Construct an AVL tree by inserting the following elements in the given order. 63, 9, 19, 27, 18, 108, 99, 81. ii)searching for a node iii)Deleting a node **Real Time Applications:** Indexing in databases, Programming Languages, Computer chess games, Computer file system, Undo function in text editor, representing city region telephone network etc. **Video link:** • https://nptel.ac.in/courses/106102064/ • http://www.digimat.in/nptel/courses/video/106106127/L50.html • https://www.youtube.com/watch?v=ffgg_zmbaxw	
UNIT-V	
Graphs: Definitions, Terminologies, Matrix and Adjacency List Representation of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search, Topological Sort. **Sorting and Searching:** Quick sort, Insertion Sort, Radix sort, Merge Sort, Address Calculation Sort. **Laboratory Sessions/ Experimental learning:** Sort a given set of elements using the sorting Method which divides input array in two halves, calls itself for the two halves and then merges the two sorted halves" and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator. **Real Time Applications:** Graph Theory, E-Commerce websites, Google Maps, Facebook **Video link:** • https://www.youtube.com/watch?v=hk5rQs7TQ7E&feature=youtu.be • **https://nptel.ac.in/courses/106/102/106102064/**	**10 Hrs**

LABORATORY EXPERIMENTS

1. A courier company has number of items to be delivered to its intended customers through its salesman. The salesman visits the following cities to deliver the respective items. Write a C program,

Sl No	Cities	Number of items
1	Agra	25
2	Chennai	50
3	Kolkata	59
4	Mumbai	72
5	Delhi	12

a) To display name of cities where salesman has delivered maximum and minimum number of items

b) To search the number of items to be delivered of a user supplied city.

2. Implement Knuth-Morris- Pratt pattern matching algorithm using C program.
3. Design, Develop and Implement a menu driven Program in C with the listed operations for the data structure which follows Last In First Out (LIFO) order. (Use Array Implementation of specified DS with maximum size MAX).
   a. Push an Element
   b. Pop an Element
   c. Demonstrate how it can be used to check Palindrome
   d. Demonstrate Overflow and Underflow situations
   e. Display the status
   f. Exit

   Support the program with appropriate functions for each of the above operations.
4. Design, Develop and Implement a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^ (Power) and alphanumeric operands.
5. Design, Develop and Implement a menu driven Program in C for the following operations on Ring Buffer of Integers (Use Array Implementation)
   a. Insert an Element on to Ring Buffer
   b. Delete an Element from Ring Buffer
   c. Demonstrate Overflow and Underflow situations on Ring Buffer
   d. Display the status of Ring Buffer
   e. Exit

   Support the program with appropriate functions for each of the above operations
6. Design, Develop and Implement a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: USN, Name, Programme, Sem, PhNo

a. Create a SLL of N Students Data by using front insertion

b. Display the status of SLL and count the number of nodes in it

c. Perform Insertion / Deletion at End of SLL

d. Perform Insertion / Deletion at Front of SLL

e. Exit

7. Design, Develop and Implement a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation, Sal, PhNo.

   a. Create a DLL of N Employees Data by using end insertion.

   b. Display the status of DLL and count the number of nodes in it.

   c. Perform Insertion and Deletion at End of DLL .

   d. Perform Insertion and Deletion at Front of DLL .

   e. Demonstrate how this DLL can be used as Double Ended Queue.

   f. Exit

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

   a) Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2.

   b) Traverse the BST recursively in inorder, pre order & post order

   Search the BST for a given element (KEY) and report the appropriate message

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

   a. Create a Graph of N cities using Adjacency Matrix.

   b. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS method

10. Develop a C program to sort a given set of n integer elements using Quick Sort method. Run the program for varied values of n and show the results of each iteration.

11. Given a File of N employee records with a set K of Keys(4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table(HT) of m memory locations with L as the set of memory addresses (2- digit) of locations in HT. Let the keys in K and addresses in L are Integers. Design and develop a Program in C that uses Hash function H: K →L as H(K)=K mod m (remainder method), and implement hashing technique to map a given key K to the address space L. Resolve the collision (if any) using linear probing.

Course Outcomes: After completing the course, the students will be able to	
CO1	Identify the necessity of data structure and its storage process.
CO2	Analyse the various operations performed on stack and queues for different applications.

CO3	Perform various operations on linked list for different applications**.**
CO4	Learn Trees and its applications.
CO5	Analyse the concepts of Graphs, searching, sorting & hashing in real time.

Reference Books	
1.	Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014.
2.	Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGraw Hill, 2014.
3.	Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Ed, Cengage Learning,2014.
4.	Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications, 2nd Ed, McGraw Hill, 2013

Continuous Internal Evaluation (CIE):

Theory for 50 Marks

CIE is executed by way of quizzes (Q), tests (T) and assignments. A minimum of three quizzes are conducted along with tests. Test portion is evaluated for 50 marks and quiz is evaluated for 10 marks. Faculty may adopt innovative methods for conducting quizzes effectively. The number of quizzes may be more than three (conduct additional quizzes and take best three). The three tests are conducted for 50 marks each and the average of all the tests are calculated for 50. The marks for the self -study are 20 (2 presentations are be held for 10 marks each). The marks obtained in test, quiz and self -studies are added to get marks out of 100 and report CIE for 50 marks.

Laboratory- 50 Marks

The laboratory session is held every week as per the time table and the performance of the student is evaluated in every session. The average of the marks over number of weeks is considered for 30 marks. At the end of the semester a test is conducted for 10 marks. The students are encouraged to implement additional innovative experiments in the lab and are awarded 10 marks. Total marks for the laboratory is 50.

Semester End Examination (SEE):

Total marks: 50+50=100

SEE for 50 marks are executed by means of an examination.

The Question paper for each course contains two parts, Part – A and Part – B. Part – A consists of objective type questions for 20 marks covering the complete syllabus. Part – B Students have to answer five questions, one from each unit for 16 marks adding up to 80 marks. Each main question may have a

maximum of three sub divisions. Each unit will have internal choice in which both questions cover entire unit having same complexity in terms of COs and Bloom's taxonomy level.

Laboratory- 50 Marks

Experiment Conduction with proper results is evaluated for 40 marks and Viva is for 10 marks. Total SEE for laboratory is 50 marks.

CO-PO/PSO Mapping														
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	-	-	-	1	-	-	-	-	-	-	2	2	-
CO2	3	3	3	-	-	-	-	-	1	-	1	2	1	-
CO3	2	2	2	1	3	-	-	-	-	-	1	3	2	3
CO4	3	2	3	-	-	-	-	-	-	2	3	2	2	-
CO5	3	2	3	-	-	-	-	-	-	2	3	2	2	2

High-3, Medium-2, Low-1

Semester: III		
ANALOG AND DIGITAL ELECTRONICS AND LAB		
Course Code: MVJ21AI35		**CIE Marks:50+50**
Credits: L:T:P: 3:0:1		**SEE Marks: 50 +50**
Hours:40 L+26P		**SEE Duration: 03+03 Hours**

Course Learning Objectives: The students will be able to	
1	Analyse the working of oscillators and use of regulators.
2	Make use of simplifying techniques in the design of combinational circuits.
3	Illustrate combinational and sequential digital circuits.
4	Demonstrate the use of flipflops and design registers and counters.
5	Design and test Analog-to-Digital and Digital-to-Analog conversion techniques.

UNIT-I	
Prerequisites : Basic analog Circuits **Metal Oxide Semiconductor Field Effect transistor(MOSFET):** Structure and I-V characteristics, MOSFET as a switch, MOSFET as an amplifier, CMOS and its applications. **Oscillators:** Basic working and applications of RC Phase shift oscillator, Wien Bridge oscillator, LC oscillator, Colpitt oscillator, Crystal Oscillator. **Linear Power Supplies:** Constituents of a Linear Power Supply, Designing Mains Transformer, Linear IC voltage regulators, Regulated Power Supply Parameters.	**8 Hrs**
UNIT-II	
Prerequisites: Digital Electronic Fundamentals **Karnaugh maps**: Minimum forms of switching functions, two and three variable Karnaugh maps, four variable karnaugh maps, Quine-McClusky Method: determination of prime implicants, The prime implicant chart, petricks method, simplification of incompletely specified functions, simplification using map-entered variables **Activity**: Writing and Analyzing C program for K-maps.	**8 Hrs**
UNIT-III	
Combinational Circuits: Multiplexer, Decoders, Adders, Subtractors, BCD arithmetic, carry look ahead adder, serial adder, ALU-Design and popular MSI chips, digital comparator, parity checker/generator, code converters, priority encoders, decoders/drivers for display devices, **Activity:** Designing a 32-bit ALU	**8 Hrs**
UNIT-IV	

Flip-Flops and Registers: **Flip Flops:** S-R,J-K,D and T flip flops,Edge-triggered JK FLIP-FLOPs **Registers:** Types of Registers, Serial In - Serial Out, Serial In - Parallel out, Parallel In - Serial Out, Parallel In - Parallel Out, Universal Shift Register, Applications of Shift Registers. **Counters:** Asynchronous Counters, Decoding Gates, Synchronous Counters, Changing the Counter Modulus, Decade Counters, Applications of Counters. **Activity:** Implementing 2 digit counters using seven segment display	**8 Hrs**

UNIT-V

D/A Conversion and A/D Conversion: **Digital to analog converters**: weighted resistor/converter, R-2R Ladder D/A converter, specifications for D/A converters, examples of D/A converter lCs, sample and hold circuit. **Analog to digital converters:** quantization and encoding, parallel comparator A/D converter, successive approximation A/D converter, counting A/D converter, dual slope A/D converter, A/D converter using voltage to frequency and voltage to time conversion, specifications of A/D converters, example of A/D Converter ICs Activity: **Demonstration of CODEC which houses both ADC and DAC.**	**8 Hrs**

LABORATORY EXPERIMENTS

1. Study of transistor phase shift oscillator and observe the effect of variation in R & C on oscillator frequency and compare with theoretical value.
2. Design and test IC 723 voltage regulator
3. Given a 4-variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.
4. Design and implement a faster way3 to add binary numbers using carry look ahead adders.
5. a) Realization and implementation of 2-bit comparator using logic gates.
   b) Implementation of 4-bit magnitude comparator using IC 7485.
6. To design and construct basic flip-flops R-S ,J-K,J-K Master slave flip-flops using gates and verify their truth table
7. Implementation of SISO, SIPO, PISO and PIPO shift registers using Flip- flops
8. Design and implementation of 3-bit synchronous up/down counter
9. Design and implement a ring counter and Johnson counter using 4-bit shift register and demonstrate its working.

10. Design and implement a mod-n (n<8) synchronous up counter using J-K Flip-Flop ICs and demonstrate its working.	
11. Design and implement an asynchronous counter using decade counter IC to count up from 0 to n (n<=9) and demonstrate on 7-segment display (using IC-7447).	
12. Design 4 bit r-2r ladder DAC using opamp.	
Course Outcomes: After completing the course, the students will be able to	
CO1	Design and analyze analog circuits using transistors, power supply, MOSFETS, regulator IC and opamp.
CO2	Simplify digital circuits using Karnaugh Map , POS and Quine-McClusky Methods
CO3	Explain construction and working of data processing circuits
CO4	Understanding the various types of latches and flip flops and building the registers and counters using flip flops.
CO5	Explain the basic principles of A/D and D/A conversion circuits and develop the same.

Reference Books	
1.	Anil K Maini, Varsha Agarwal, Electronic Devices and Circuits, Wiley, 2012.
2.	Charles H Roth and Larry L Kinney, Fundamentals of Logic design, Cengage Learning,2019.
3.	M. Morris Mani, Digital Design, 4th Edition, Pearson Prentice Hall, 2008.
4.	David A. Bell, Electronic Devices and Circuits, 5th Edition, Oxford University Press, 2008

Continuous Internal Evaluation (CIE):

Theory for 50 Marks

CIE is executed by way of quizzes (Q), tests (T) and assignments. A minimum of three quizzes are conducted along with tests. Test portion is evaluated for 50 marks and quiz is evaluated for 10 marks. Faculty may adopt innovative methods for conducting quizzes effectively. The number of quizzes may be more than three (conduct additional quizzes and take best three). The three tests are conducted for 50 marks each and the average of all the tests are calculated for 50. The marks for the assignments are 20 (2 assignments for 10 marks each).The marks obtained in test, quiz and assignment are added to get marks out of 100 and report CIE for 50 marks.

Laboratory- 50 Marks

The laboratory session is held every week as per the time table and the performance of the student is evaluated in every session. The average of the marks over number of weeks is considered for 30 marks. At the end of the semester a test is conducted for10 marks. The students are encouraged to implement

additional innovative experiments in the lab and are awarded 10 marks. Total marks for the laboratory is 50.

Semester End Examination (SEE):

Total marks: 50+50=100

SEE for 50 marks are executed by means of an examination.

The Question paper for each course contains two parts, Part – A and Part – B. Part – A consists of objective type questions for 20 marks covering the entire syllabus. Part – B Students have to answer five questions, one from each unit for 16 marks adding up to 80 marks. Each main question may have a maximum of three sub divisions. Each unit will have internal choice in which both questions cover entire unit having same complexity in terms of COs and Bloom's taxonomy level.

CO-PO/PSO Mapping														
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	**3**	**3**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**2**	**-**
CO2	**3**	**3**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**2**	**-**
CO3	**3**	**3**	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**2**	**-**
CO4	**3**	**3**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**2**	**-**
CO5	**3**	**3**	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1**	**1**	**2**

High-3, Medium-2, Low-1

Semester: III		
UNIX SHELL PROGRAMMING (Theory)		
Course Code: MVJ21AEC37		**CIE Marks:100**
Credits: L:T:P:S:2:0:0:0		**SEE Marks: 100**
Hours: 40L		**SEE Duration: 3 Hrs**

Course Learning Objectives: The students will be able to	
1	To help the students to understand effective use of Unix concepts, commands and terminology
2	Identify, access, and evaluate UNIX file system.
3	Understand UNIX command syntax and semantics.
4	Ability to read and understand specifications, scripts and programs.

UNIT-I	
Introduction of UNIX - Introduction, History, Architecture, Experience the Unix environment, Basic commands ls, cat, cal, date, calendar, who, printf, tty, sty, uname, passwd, echo, tput, and bc.	**6 Hrs**
UNIT-II	
UNIX File System- The file, what's in a filename? The parent-child relationship, pwd, the Home directory, absolute pathnames, using absolute pathnames for a command, cd, mkdir, rmdir, Relative pathnames, The UNIX file system.	**6 Hrs**
UNIT-III	
Basic File Attributes - Is – l, the –d option, File Permissions, chmod, Security and File Permission, users and groups, security level, changing permission, user masks, changing ownership and group, File Attributes, More file attributes: hard link, symbolic link, umask, find.	**6 Hrs**
UNIT-IV	
Introduction to the Shell Scripting - Introduction to Shell Scripting, Shell Scripts, read, Command Line Arguments, Exit Status of a Command, The Logical Operators && and ||, exit, if, and case conditions, expr, sleep and wait, while, until, for, $, @, redirection. The here document, set, trap, Sample Validation and Data Entry Scripts.	**6 Hrs**
UNIT-V	
Introduction to UNIX System process: Mechanism of process creation. Parent and child process. The ps command with its options. Executing a command at a specified point of time: at command. Executing a command periodically: cron command and the crontab file Signals.	**6 Hrs**

Course Outcomes: After completing the course, the students will be able to	
CO1	Know the basics of Unix concepts and commands.
CO2	Evaluate the UNIX file system.
CO3	Apply Changes in file system.
CO4	Understand scripts and programs.
CO5	Analyze Facility with UNIX system process

Reference Books	
1.	Unix Concepts & Applications 4rth Edition, Sumitabha Das, Tata McGraw Hill
2.	Unix Shell Programming, Yashwant Kanetkar
3.	Introduction to UNIX by M G Venkatesh Murthy

Continuous Internal Evaluation (CIE):

Theory for 50 Marks

CIE is executed by way of quizzes (Q), tests (T) and assignments. A minimum of three quizzes are conducted along with tests. Test portion is evaluated for 50 marks and quiz is evaluated for 10 marks. Faculty may adopt innovative methods for conducting quizzes effectively. The number of quizzes may be more than three (conduct additional quizzes and take best three). The three tests are conducted for 50 marks each and the average of all the tests are calculated for 50. The marks for the assignments are 20 (2 assignments for 10 marks each). The marks obtained in test, quiz and assignment are added to get marks out of 100 and report CIE for 50 marks.

Semester End Examination (SEE):

Total marks: 50+50=100

SEE for 50 marks is executed by means of an examination. The Question paper for each course contains two parts, Part – A and Part – B. Part – A consists of objective type questions for 20 marks covering the entire syllabus. Part – B Students have to answer five questions, one from each unit for 16 marks adding up to 80 marks. Each main question may have a maximum of three sub divisions. Each unit will have internal choice in which both questions cover entire unit having same complexity in terms of COs and Bloom's taxonomy level.

CO-PO Mapping												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	**2**	**2**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
CO2	**2**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
CO3	**3**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
CO4	**3**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
CO5	**3**	**2**	**3**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**

High-3, Medium-2, Low-1

Semester: III		
Additional Mathematics-I **(Common to all branches)**		
Course Code:	**MVJ21MATDIP1**	**CIE Marks:50**
Credits:	**L:T:P:S: 4:0:0:0**	**SEE Marks: 50**
Hours:	**40L**	**SEE Duration: 3 Hrs**

Course Learning Objectives: The students will be able to	
1	To familiarize the important and introductory concepts of Differential calculus
2	Aims to provide essential concepts integral calculus
3	To gain knowledge of vector differentiation
4	To learn basic study of probability
5	Ordinary differential equations of first orderand analyze the engineering problems.

UNIT-I	
Differential calculus: Recapitulation of successive differentiation -nth derivative -Leibnitz theorem (without proof) and Problems, Polar curves - angle between the radius vector and tangent, angle between two curves, pedal equation, Taylor's and Maclaurin's series expansions- Illustrative examples. **Video Link**: 1. http://nptel.ac.in/courses.php?disciplineID=111	**8 Hrs**
UNIT-II	
IntegralCalculus: Statement of reduction formulae for the integrals of $\sin^n(x)$, $\cos^n(x)$, $\sin^n(x)\cos^n(n)$ and evaluation of these integrals with standard limits-problems. Double and triple integrals-Simple examples. **Video Link**: 1. http://nptel.ac.in/courses.php?disciplineID=111	**8 Hrs**
UNIT-III	
Vector Differentiation: ScalarandVectorpointfunctions,Gradient,Divergence,Curl,SolenoidalandIrrotationalvectorfields. **Vector identities**- $div(\phi\vec{A})$, $curl(\phi\vec{A})$, $curl(grad(\phi))$, $div(curl\,\vec{A})$. **Video Link**: 1. http://nptel.ac.in/courses.php?disciplineID=111	**8Hrs**
UNIT-IV	

Probability: Basic terminology, Sample space and events. Axioms of probability. Conditional probability – illustrative examples. Bayes theorem-examples. **Video Link**: 1. http://nptel.ac.in/courses.php?disciplineID=111	**8Hrs**
UNIT-V	
Ordinary Differential Equations of First Order: Introduction – Formation of differential equation, solutions of first order and first degree differential equations: variable separable form, homogeneous, exact, linear differential equations. **Video Link**: 1. http://nptel.ac.in/courses.php?disciplineID=111	**8Hrs**

Course Outcomes: After completing the course, the students will be able to	
CO1	Apply the knowledge of calculus to solve problems related to polar curves and its applications
CO2	Apply the concept of integration and variables to evaluate multiple integrals and their usage in computing the area and volumes.
CO3	Illustrate the applications of multivariate calculus to understand the solenoidal and irrotational vectors and also exhibit the inter dependence of line, surface and volume integrals.
CO4	Understand the basic Concepts of Probability
CO5	Recognize and solve first-order ordinary differential equations occurring in different branches of engineering.

Reference Books	
1.	B.S. Grewal,Higher Engineering Mathematics**,** Khanna Publishers,43rd Edition**,** 2013**,** .
2.	G. B. Gururajachar, Calculus and Linear Algebra, Academic Excellent Series Publication, 2018-19
3.	Chandrashekar K. S, Engineering Mathematics-I, Sudha Publications, 2010.

Continuous Internal Evaluation (CIE):

Theory for 50 Marks

CIE is executed by way of quizzes (Q), tests (T) and assignments. A minimum of three quizzes are conducted along with tests. Test portion is evaluated for 50 marks and quiz is evaluated for 10 marks. Faculty may adopt innovative methods for conducting quizzes effectively. The number of quizzes may be more than three (conduct additional quizzes and take best three). The three tests are conducted for 50 marks each and the average of all the tests are calculated for 50. The marks for the assignments are 20 (2 assignments for 10 marks each). The marks obtained in test, quiz and assignment are added to get marks out of 100 and report CIE for 50 marks.

Semester End Examination (SEE):

Total marks: 50+50=100

SEE for 50 marks is executed by means of an examination. The Question paper for each course contains two parts, Part – A and Part – B. Part – A consists of objective type questions for 20 marks covering the entire syllabus. Part – B Students have to answer five questions, one from each unit for 16 marks adding up to 80 marks. Each main question may have a maximum of three sub divisions. Each unit will have internal choice in which both questions cover entire unit having same complexity in terms of COs and Bloom's taxonomy level.

CO-PO Mapping												
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	3	0	2	0	0	0	0	0	0	1	1
CO2	3	3	0	2	0	0	0	0	0	0	1	1
CO3	3	3	0	3	0	0	0	0	0	0	1	1
CO4	2	2	0	3	0	0	0	0	0	0	1	1
CO5	2	2	0	2	0	0	0	0	0	0	0	1

High-3, Medium-2, Low-1