

**MRSPTU UNDER GRADUATE OPEN ELECTIVES MODIFICATIONS
(2020 BATCH ONWARDS)**

Modified list of UG Open Electives offered by Deptt. of CSE

Column A			Column B		
Previously uploaded on 26.08.2020			Modified for 2K20 Batch onwards		
Subject Name	Subject Code	L – T – P – C	Subject Name	Subject Code	L – T – P – C
Data Structures & Algorithms	BCSEO1-001	3 – 0 – 0 – 3	Data Structures & Algorithms	BCSEO1-001	2 – 0 – 0 – 2
			Data Structures & Algorithms laboratory	BCSEO1-051	0 – 0 – 2 – 1
MATLAB Programming	BCSEO1-002	3 – 0 – 0 – 3	Computational Sciences and Problem Solving	BCSEO1-002	2 – 0 – 0 – 2
			Computational Sciences and Problem Solving Lab	BCSEO1-052	0 – 0 – 2 – 1
Object Oriented Programming	BCSEO1-014	3 – 0 – 0 – 3	Object Oriented Programming	BCSEO1-014	2 – 0 – 0 – 2
			Object Oriented Programming Laboratory	BCSEO1-053	0 – 0 – 2 – 1

Note:

1. The previously uploaded open electives of CSE Deptt. (Column A), stands modified as given in Column B for 2K20 Batch onwards.
2. The other open electives offered by Department of CSE (except Column A) previously uploaded on 26.08.2020 will remain same for all the batches.
3. Only subject name MATLAB Programming has been replaced by Computational Sciences and Problem Solving from 2K20 Batch onwards. However, syllabi of both the subjects are identical.

**MRSPTU UNDER GRADUATE OPEN ELECTIVES MODIFICATIONS
(2020 BATCH ONWARDS)**

DATA STRUCTURES AND ALGORITHMS

Subject Code- BCSE01-001

L T P C
2 0 0 2

Duration – 30 Hrs

COURSE OBJECTIVE:

1. To impart the basic concepts of data structures and algorithms
2. To understand concepts about searching and sorting techniques.
3. To understand basic concepts about stacks, queues, lists, trees and graphs

COURSE OUTCOMES

CO1: For a given algorithm student will be able to analyze the algorithms and justify the correctness.

CO2: To learn basics of stacks and queues.

CO3: To learn linked list concepts.

CO4: To learn different sorting algorithms.

COURSE CONTENTS

UNIT-I (08 hrs)

Introduction: Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc. Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off. Searching: Linear Search and Binary Search Techniques.

UNIT-II (07 hrs)

Stacks and Queues: ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks, Queues and its types.

UNIT-III (07 hrs)

Linked Lists: Introduction to Linked Lists and its types, Representation of single linked list in memory, operations: Traversing, Searching, Insertion into, Deletion from linked list; Trees: Basic Tree Terminologies, Different types of Trees: Binary Tree, Binary Search trees

UNIT-IV (08 hrs)

Sorting and Hashing: Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Performance and comparison among all the methods, Hashing. Graph: Basic Terminologies and Representations.

RECOMMENDED BOOKS:

1. Fundamentals of Data Structures, Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press.
2. Algorithms, Data Structures, and Problem Solving with C++, Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company
3. —How to Solve it by Computer, 2nd Impression by R.G. Dromey, Pearson Education.

**MRSPTU UNDER GRADUATE OPEN ELECTIVES MODIFICATIONS
(2020 BATCH ONWARDS)**

DATA STRUCTURES & ALGORITHMS LABORATORY

Subject Code- BCSE01-051

**L T P C
0 0 2 1**

COURSE OUTCOMES:

CO1 To introduce the basic concepts of Data structure, basic data types, searching and sorting based on array data types.

CO2 To introduce the structured data types like Stacks and Queue and its basic operation's implementation

CO3 To introduces dynamic implementation of linked list

CO4 To introduce the concepts of Tree and graph and implementation of traversal algorithms

PRACTICALS:

1. Write a program for Linear search methods.
2. Write a program for Binary search methods.
3. Write a program for insertion sort, selection sort and bubble sort.
4. Write a program to implement Stack and its operation.
5. Write a program for quick sort.
6. Write a program to implement Queue and its operation.
7. Write a program to implement singly linked list for the following operations: Create, Display, searching, traversing and deletion.

**MRSPTU UNDER GRADUATE OPEN ELECTIVES MODIFICATIONS
(2020 BATCH ONWARDS)**

COMPUTATIONAL SCIENCES AND PROBLEM SOLVING

Subject Code- BCSE01-002

L T P C
2 0 0 2

Duration- 30 Hrs

COURSE OBJECTIVE:

1. To learn the MATLAB environment and its programming fundamentals
2. Ability to write Programs using commands and functions

COURSE OUTCOMES:

CO1: Student will able to use MATLAB environment for writing, debugging and executing programs.

CO2: To be able to learn different operators used in MATLAB.

CO3: To learn how to use data types.

CO4: To learn 1D and 2D plotting.

COURSE CONTENTS

UNIT I (08 Hrs)

Introduction to MATLAB Software: MATLAB window, Command window, Workspace, Command history, Setting directory, working with the MATLAB user interface, Basic commands, Writing Script file, Executing script files The MATLAB Editor, Saving m files.

UNIT-II (07 Hrs)

Assigning variable, operation with variables.BODMAS Rules, Arithmetic operations, Operators and special characters, Mathematical and logical operators, Solving arithmetic equations

UNIT III (07 Hrs)

Data files and Data types: Character and string, Arrays and vectors, Column vectors and Row vectors

Creating rows and columns Matrix, Matrix operations, finding transpose, determinant and Solving matrices

UNIT IV (08 Hrs)

1D and 2D Plotting: Plotting vector and matrix data, Plot labelling, curve labelling and editing, 2D plots: Basic Plotting Functions, Creating a Plot, Plotting Multiple Data Sets in One Graph Specifying Line Styles and Colors Graphing, Displaying Multiple Plots in One Figure, Controlling the Axes.

RECOMMENDED BOOKS

1. "MATLAB and its applications in Engineering", by R.K. Bansal, A.K. Goel and M.K Sharma, PEARSON
2. "MATLAB: Easy way of Learning"by S. Swapna Kumar and Lenina S. V. B.

**MRSPTU UNDER GRADUATE OPEN ELECTIVES MODIFICATIONS
(2020 BATCH ONWARDS)**

COMPUTATIONAL SCIENCES AND PROBLEM SOLVING LABORATORY

Subject Code- BCSE01-052

L T P C

0 0 2 1

COURSE OUTCOMES:

CO1 Students will be able to learn the introduction to MATLAB.

CO2 To be able to learn about inbuilt functions.

CO3 To be able to learn various matrix operations.

CO4 To be able to learn various control statements.

PRACTICALS:

1. Introduction to MATLAB environment and types of MATLAB files.
2. Use of help command to get help about different inbuilt functions.
3. Write a program to show the output of various unary and binary operators.
4. Write programs for Matrix Manipulations, (reshaping matrices, expanding matrix size, appending or deleting a row/column to a matrix, concatenation of matrices).
5. Write programs which demonstrate the use special matrices.
6. Write programs to show output for various matrix and array operations.
7. Write programs for demonstrating the use for various control statements.
8. Write a MATLAB code for computing factorial of a number n. Assume n is already defined. The code should return a scalar, not a vector.

**MRSPTU UNDER GRADUATE OPEN ELECTIVES MODIFICATIONS
(2020 BATCH ONWARDS)**

OBJECT ORIENTED PROGRAMMING

Subject Code- BCSE01-014

**L T P C
2 0 0 2**

Duration – 30 hrs.

COURSE OBJECTIVE

To introduce the principles and paradigms of Object Oriented Programming Language for design and implement the Object Oriented System

COURSE OUTCOMES

CO1 To introduce the basic concepts of object oriented programming language and its representation

CO2 To understand the concept of memory allocation.

CO3 To introduce polymorphism and overloading of operator.

CO4 To learn the concept of text streams.

COURSE CONTENT

UNIT-I (08 hrs)

Introduction to C++, C++ Standard Library, Illustrative Simple C++ Programs. Header Files, Namespaces, Application of object oriented programming. Object Oriented Concepts, Introduction to Objects and Object Oriented Programming, Encapsulation, Polymorphism, Overloading, Inheritance, Abstract Classes, Access specifier (public/ protected/ private), Class Scope.

UNIT-II (08 hrs)

This Pointer, Dynamic Memory Allocation and De-allocation (New and Delete), Static Class Members, Constructors, parameter Constructors and Copy Constructors, Destructors, Introduction of inheritance, Types of Inheritance

UNIT-III (07 hrs)

Polymorphism and its types, Fundamentals of Operator Overloading, Rules for Operators Overloading, Implementation of Overloading Unary Operators, Binary Operators.

UNIT-IV (07 hrs)

Text Streams and binary stream, Sequential and Random Access File, Stream Input/ Output Classes, Stream Manipulators.

RECOMMENDED BOOKS:

1. Robert Lafore, _Object Oriented Programming in Turbo C++_, 2nd Ed., The WAITE Group Press, 1994.
2. Herbert shield, _The complete reference C ++_, 4th Ed., Tata McGraw Hill, 2003.
3. Shukla, _Object Oriented Programming in C++_, Wiley India, 2008.
4. H M Deitel and P J Deitel, _C++ How to Program_, 2nd Ed., Prentice Hall, 1998.
5. D Ravichandran, _Programming with C++_, 3 rd Ed., Tata McGraw Hill, 2003.
6. Bjarne Stroustrup, _The C++ Programming Language_, 4th Ed., Addison Wesley, 2013.
7. R. S. Salaria, _Mastering Object-Oriented Programming with C++_, Salaria Publishing House, 2016.

OBJECT ORIENTED PROGRAMMING LABORATORY

Subject Code- BCSE01-053

**L T P C
0 0 2 1**

COURSE OUTCOMES:

CO1 To learn about classes and objects.

CO2 To implement Constructors and Destructors

CO3 To learn Operator Overloading

CO4 To implement typecasting

PRACTICALS:

1. Classes and Objects- Write a program that uses a class where the member functions are defined inside a class.
2. Classes and Objects- Write a program that uses a class where the member functions are defined outside a class.
3. Classes and Objects- Write a program to demonstrate the use of static data members.
4. Classes and Objects- Write a program to demonstrate the use of const data members.
5. Constructors and Destructors- Write a program to demonstrate the use of zero argument and parameterized constructors.
6. Operator Overloading- Write a program to demonstrate the overloading of increment and decrement operators.
7. Operator Overloading- Write a program to demonstrate the overloading of binary arithmetic operators.
8. Typecasting- Write a program to demonstrate the typecasting of basic type to class type.
9. Typecasting- Write a program to demonstrate the typecasting of class type to basic type.