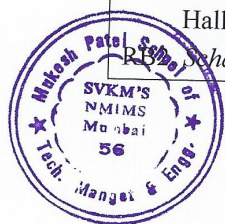


**SVKM's NMIMS**  
**Mukesh Patel School of Technology Management & Engineering**

<b>Program:</b> B. Tech. ( I.T., Computer, EXTC, Mechanical, Civil, Mechatronics & Data Science)					<b>Semester:</b> II	
<b>Course/Module:</b> Programming for Problem Solving					<b>Module Code:</b> BTIT02009, BTCO02009, BTET02009, BTME02009, BTCIO2009, BTMA02009, BTDS02009	
<b>Teaching Scheme</b>					<b>Evaluation Scheme</b>	
<b>Classroom Session</b>	<b>Lecture (Hours per week)</b>	<b>Tutorial (Hours per week)</b>	<b>Practical/ Group work (Hours per week)</b>	<b>Credit</b>	<b>Continuous Evaluation (Marks-50)</b>	<b>Term End Examinations (TEE) (Marks - 100 in Question Paper)</b>
42	3	0	4	5	Marks Scaled to 50	Marks Scaled to 50
<b>Course Rationale:</b> This course aims to teach the fundamental concepts of Procedural Programming. Students will develop skills related to problem solving by writing computer programs. This course does not require any prior programming experience.						
<b>Course Objectives:</b> <ol style="list-style-type: none"> <li>1. To enable students understand the basic concepts of Programming and help them build Programming Logic.</li> <li>2. To develop problem solving skills using basic Programming constructs, Decision Making and Looping.</li> <li>3. To enable students solve complex problems using the knowledge of Arrays, Functions, Structures and Pointers.</li> </ol>						
<b>Course Outcomes:</b> After completion of the course, students would be able to : <ol style="list-style-type: none"> <li>1. formulate algorithms and draw flowcharts for arithmetic and logical problems</li> <li>2. implement Decision Making, Nested Control Structures and Iterations</li> <li>3. implement programs using Functions and concept of Recursion</li> <li>4. demonstrate the use of Arrays, Strings, Structures and Pointers</li> <li>5. apply Programming knowledge to solve searching and sorting problems</li> </ol>						
<b>Pedagogy:</b> Peer learning, Group exercises, quizzes, presentations and lecture method						
<b>Textbooks:</b> TB1. <i>Schaum's Outline Programming with C</i> , 3 e, Byron Gottfried, McGraw-Hill, 2017. TB2. <i>Programming in ANSI C</i> , 7 e, E. Balaguruswamy, Tata McGraw Hill Education, 2017.						
<b>Reference Books:</b> RB1. <i>The C Programming Language</i> , 2 e, Brian W. Kernighan and Dennis M. Ritchie, Prentice Hall of India, 1988. RB2. <i>Schaum's Outlines Data Structures</i> , Revised 1 e, Seymour Lipschutz, Tata McGraw Hill, 2014.						



B. Tech.  
17/7/2018

**SVKM's NMIMS**  
**Mukesh Patel School of Technology Management & Engineering**

**Links to websites:**

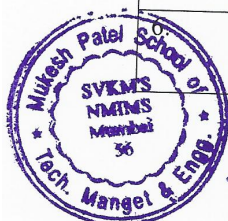
- <https://cprogrammingcodes.blogspot.in/2011/09/algorithms-and-flowchart.html>

**Evaluation Scheme:**

• Tutorial Test/Presentation/viva/quiz	20%
• Mid Term	30%
• Term End Exam	50%
<b>Total</b>	<b>100%</b>

**Session Plan:**

Session	Topics	Session\ Course Outcomes	Pedagogical Tool	Textbook Chapters & Readings
<b>Unit 1</b>	<b>Introduction to Programming</b>	<b>CO1</b>	--	--
1.	Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.)	Formulate algorithms	Lectures	<b>TB1:</b> Chapter 1: Introductory Concepts
2.	<b>Idea of Algorithm:</b> <ul style="list-style-type: none"> <li>Steps to solve logical and numerical problems.</li> </ul> <b>Representation of Algorithm:</b> <ul style="list-style-type: none"> <li>Flowchart/Pseudo code with examples</li> </ul>	<ul style="list-style-type: none"> <li>Formulate algorithms</li> <li>Draw flowcharts for arithmetic and logical problems</li> </ul>		<b>TB1:</b> Chapter 1: Introductory Concepts
3.	<b>From algorithms to programs:</b> <ul style="list-style-type: none"> <li>Source code, variables (with data types) variables and memory locations, Syntax and Logical Errors in compilation, object and executable code</li> </ul>	Formulate algorithms		<b>TB1:</b> Chapter 2: Introduction to C programming <b>TB2:</b> Chapter 1: Overview of C Chapter 2: Constants variables and data types
<b>Unit 2</b>	<b>Basic Programming Constructs</b>	<b>CO2</b>	--	--
4.	<b>Operators</b>	Evaluate arithmetic and logical operator	<ul style="list-style-type: none"> <li>Lectures</li> <li>Presentations</li> <li>Quizzes</li> </ul>	<b>TB1, TB2:</b> Chapter 3: Operators and expressions <b>TB1:</b> Chapter 5: Control
5.	<b>Expressions</b>	Evaluate arithmetic and logical expressions		
	<b>Decision making and Branching:</b> <ul style="list-style-type: none"> <li>If statements and if else statement</li> </ul>	Implement Decision Making		



*[Signature]*  
17/7/2018



**SVKM's NMIMS**  
**Mukesh Patel School of Technology Management & Engineering**

7.	<b>Nesting of if else statements and Else if ladder</b>	Implement Decision Making		Statements <b>TB2:</b> Chapter 5: Decision making and branching
8.	<b>Switch statements</b>	Implement Decision Making and nested control		
9.	<b>Continue statement Break statement</b>	Implement Decision Making and nested control		
10.	<b>Looping – while</b>	Implement iteration loop		<b>TB1:</b> Chapter 6: Control Statements
11.	<b>do-while</b>	Implement iteration loop		
12.	<b>For loops</b>	Implement iteration loop		<b>TB2:</b> Chapter 6: Decision making and looping
13.	<b>For loops and Finding roots of equations</b>	Implement iteration loop and iterations		
14.	<b>Nested loops</b>	Nested Control Structures and iterations		
<b>Unit 3</b>	<b>Arrays and Strings</b>	<b>CO4</b>	--	--
15.	<ul style="list-style-type: none"> <li>Concept, declaration, initialization</li> <li>Accessing array elements of one-dimensional array</li> </ul>	Demonstrate the use of Arrays	<ul style="list-style-type: none"> <li>Lectures</li> <li>Problem Solving</li> <li>Quizzes</li> </ul>	<b>TB1:</b> Chapter 9: Arrays
16.	<b>One-dimensional array</b>	Demonstrate the use of Arrays		<b>TB2:</b> Chapter 7: Arrays
17.	<ul style="list-style-type: none"> <li>Concept, declaration, initialization</li> <li>Accessing array elements of two-dimensional array</li> </ul>	Demonstrate the use of Arrays		
18.	<b>Two - dimensional array</b>	Demonstrate the use of Arrays		
19.	<b>Introduction to strings</b>	Demonstrate the use of strings		
<b>Unit 4</b>	<b>Functions</b>	<b>CO3</b>	--	--
20.	<b>Function</b> <ul style="list-style-type: none"> <li>Introduction and need of user defined functions</li> </ul>	Implement programs using Functions	<ul style="list-style-type: none"> <li>Lectures</li> <li>Problem Solving</li> <li>Quizzes</li> </ul>	<b>TB1:</b> Chapter 7: Functions
21.	<ul style="list-style-type: none"> <li>Defining a Function</li> <li>Function calls and declaration</li> </ul>	Implement programs using Functions		<b>TB2:</b> Chapter 9: User defined functions
22.	<b>Category of functions:</b> <ul style="list-style-type: none"> <li>No argument and no return value</li> <li>Argument but no return value</li> </ul>	Implement programs using Functions		

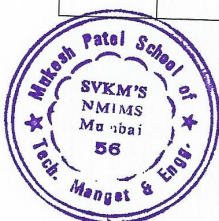


B.P.

17/11/2018

**SVKM's NMIMS**  
**Mukesh Patel School of Technology Management & Engineering**

23.	<b>Category of functions:</b> <ul style="list-style-type: none"> <li>Argument with return value</li> <li>No argument but return value</li> </ul>	Implement programs using Functions		
24.	<b>Passing arrays to functions</b>	Implement programs using Functions		<b>TB1:</b> Chapter 9: Arrays <b>TB2:</b> Chapter 9: User defined functions, Character arrays and strings
25.	Declaring & initialising string variable, Reading & writing strings	Demonstrate the use Strings		
26.	String handling functions	Demonstrate the use Strings		
27.	Passing strings to functions	Demonstrate the use Strings		
<b>Unit 5</b>	<b>Recursion</b>	<b>CO3</b>	--	--
28.	<b>Introduction to Recursion</b> <ul style="list-style-type: none"> <li>Recursion as a different way of solving problems adjoint method</li> </ul>	Implement programs using concept of Recursion	<ul style="list-style-type: none"> <li>Lectures</li> <li>Problem Solving</li> <li>Presentation</li> </ul>	<b>TB1:</b> Chapter 7: Functions <b>TB2:</b> Chapter 9: User defined functions
29.	<b>Recursion programs:</b> <ul style="list-style-type: none"> <li>Examples - Finding Factorial, Fibonacci series</li> </ul>	Implement programs using concept of Recursion		
30.	<b>Recursion programs:</b> <ul style="list-style-type: none"> <li>Examples GCD, Merge sort.</li> </ul>	Implement programs using concept of Recursion		
<b>Unit 6</b>	<b>Structures</b>	<b>CO4</b>	--	--
31.	<ul style="list-style-type: none"> <li>Defining a Structure Declaring structure variables</li> <li>Accessing structure members</li> <li>Structure Initialization</li> </ul>	Demonstrate the use of structures	<ul style="list-style-type: none"> <li>Lectures</li> <li>Problem Solving</li> <li>Quizzes</li> <li>Presentation</li> </ul>	<b>TB1:</b> Chapter 11: Structure and unions <b>TB2:</b> Chapter 10: Structure and unions
32.	Array of Structure	Demonstrate the use of structures		
33.	Structure within structure	Demonstrate the use of structures		
34.	Difference between Structure and Unions	Demonstrate the use of structures		
<b>Unit 7</b>	<b>Pointers</b>	<b>CO4</b>	--	--
35.	<ul style="list-style-type: none"> <li>Idea of pointers</li> <li>Defining pointers</li> </ul>	Demonstrate the use of pointers	<ul style="list-style-type: none"> <li>Lectures</li> <li>Problem Solving</li> <li>Quizzes</li> <li>Presentation</li> </ul>	<b>TB1:</b> Chapter 10: Pointers <b>TB2:</b> Chapter 11: Pointers
36.	Call by value and call by reference	Demonstrate the use of pointers		



**SVKM's NMIMS**  
**Mukesh Patel School of Technology Management & Engineering**

37.	Use of Pointers in self-referential structures, Notion of linked list (no implementation)	Demonstrate the use of pointers	n	<b>TB1:</b> Chapter 11: Structure and unions <b>TB2:</b> Chapter 13: Dynamic memory allocation and linked
<b>Unit 8</b>	<b>Basic Algorithms</b>	<b>CO5</b>	--	--
38.	Notion of order of complexity through example programs (no formal definition required)	Apply Programming knowledge to solve searching and sorting problems	<ul style="list-style-type: none"> <li>• Lectures</li> <li>• Problem Solving</li> <li>• Quizzes</li> <li>• Presentation</li> </ul>	<b>RB2:</b> Chapter 9: Sorting and searching
39.	Searching: Sequential search	Apply Programming knowledge to solve searching		
40.	Basic Sorting Algorithms : Bubble sort	Apply Programming knowledge to solve sorting problems		
41.	Insertion sort	Apply Programming knowledge to solve sorting problems		
42.	Selection sort	Apply Programming knowledge to solve sorting problems		

Signature  
(Prepared by Concerned Faculty/HOD)



Signature  
(Approved by Dean)



**SVKM's NMIMS**  
**Mukesh Patel School of Technology Management & Engineering**

<b>Program: B. Tech. -Data Science (Business Analytics)</b>				<b>Semester : III</b>	
<b>Course/Module: Data Structure and Algorithm</b>				<b>Module Code: BTDS03002</b>	
<b>Teaching Scheme</b>				<b>Evaluation Scheme</b>	
<b>Lecture (Hours per week)</b>	<b>Practical (Hours per week)</b>	<b>Tutorial (Hours per week)</b>	<b>Credit</b>	<b>Internal Continuous Assessment (ICA) (Marks -50)</b>	<b>Term End Examinations (TEE) (Marks -100 in Question Paper)</b>
3	4	0	5	Marks Scaled to 50	Marks Scaled to 50
<b>Pre-requisite:</b> Computer Programming – I (basic programming skills)					
<b>Objectives:</b> <ul style="list-style-type: none"> <li>To provide knowledge of data structure and its type</li> <li>To provide advanced computer Science Programming of different data structure background for complex programming skill</li> </ul>					
<b>Outcomes:</b> After completion of the course, students would be able to : <ul style="list-style-type: none"> <li>distinguish data and information</li> <li>learn about data structure</li> <li>define various types of data structures</li> <li>know different data structure operations</li> <li>describe about data types in C</li> <li>define abstract data types</li> </ul>					
<b>Detailed Syllabus: (per session plan)</b>					
<b>Unit</b>	<b>Description</b>				<b>Duration</b>
1	<b>Introduction to Data Structures:</b> Introduction to the theory of data structure, classification of data structure, Data representation, Abstract data types with examples, Data types Recursion: Recursive definition, recursion to solve the real life problems, Importance of data structures, Fundamental of operations of various data structures such as array, Structure etc.				4
2	<b>Array, stack and Queues"</b> Array, Stack and queue representation, Operation and examples, Polish notation, stack in recursion Queue, Queue processing, Types of Queues: Circular queues and priority queues, Role of data structure in optimized coding				10
3	<b>List and Linked lists:</b> Linked list and array, Dynamic linked lists (single, doubly, circular) – processing, operations, applications of linked lists				7
4	<b>Tree:</b> Concept of trees, Binary tree and its representation, complete binary tree, tree traversal algorithms, in-order, preorder, post-order, Height balanced tree, Threaded binary tree, Expression tree and game tree, Huffman algorithm and its applications Search Trees: Binary search tree, insertion and deletion, AVL tree, M-way search tree, B tree and B+ tree (including insertion and deletion). Applications of tree data structures to solve real life problems				10



**Signature**  
 (Prepared by Concerned Faculty/HOD)

**SVKM's NMIMS**  
**Mukesh Patel School of Technology Management & Engineering**

5	<b>Graph:</b> Terminology, sequential and linked representation of graphs, adjacency matrices, Adjacency list, adjacency multi list, graph traversal: breadth first search and Depth first search, connected component, spanning trees, minimum cost spanning tree, Prims and Kruskal algorithms, transitive closure and shortest path algorithm, Application of graph to solve the real life problems.	6
6	<b>Searching, Sorting and Hashing:</b> General background, Basic search techniques, sequential searching, bubble sort, selection sort, insertion sort, Shell sort and Radix sort and their efficiency, Quick sort and merge sort comparison, indexed sequential searching, binary search, Hashing: Basics, collision resolution and their efficiency comparison, Application of search and indexes to solve real life problems.	8
<b>Total</b>		45

**Text Books:**

1. C and Data Structure, P.S. Deshpande and O.G. Kakde, CHARLES RIVER MEDIA, INC. Hingham, Massachusetts (pdf version)
2. Reema Thareja, "Data Structure using C", Oxford University Press, 2<sup>nd</sup> Edition, 2014
3. Y. Langsam, M.J. Augenstein, A.M. Tenenbaum; "Data structure using C and C++", 2<sup>nd</sup> Edition, PHI2004
4. Seymour Lipschutz, "Data Structures", Schaum's Outlines, Tata McGraw Hill, 2006

**Reference Books:**

1. Richard F. Gillberg, Behrouz A. Forouzan, "Data Structure – A Pseudo Approach with C", Cengage Publication, 2005
2. G.S. Baluja, "Data Structure through C (A practical approach)", Dhanpat rai and co. 4<sup>th</sup> edition, 2014

**Any other information: NIL**

**Total Marks of Internal Continuous Assessment (ICA): 50 Marks**

**Distribution of ICA Marks:**

Description of ICA	Marks
Test Marks	20
Term Work Marks	30
<b>Total Marks :</b>	<b>50</b>

**Details of Term work:**

1. Minimum: Practical based on 10 Experiments
2. Minimum Two class tests.
3. Minimum two assignments



**Signature**  
**(Prepared by Concerned Faculty/HOD)**