GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Integrated MCA 2nd Semester
Subject Name: Advanced C Programming (Adv. C)
Subject Code: 2628602

With effective from academic year 2018-19

1. Objectives:

- To be able to understand and use pointers in C Programs.
- To be able to create user defined data types in C
- To be able to write C application which can do input/output on files.

2. Prerequisites: Basic knowledge of C Programming

3. Course Contents:

Unit No.	Course Content	Weightage Percentage
1	**Call By Value, Call by Reference, Recursion** **Pointers** Introduction, Understanding Memory Addresses, Address Operator (&), Pointers (Declaring a Pointer, Initializing Pointers, Indirection Operator and Dereferencing, void Pointer, Null Pointer, Use of Pointers), Arrays and Pointers (One-dimensional Arrays and Pointers, Passing an Array to a Function, Differences between Array Name and Pointer), Pointer and String, Pointer Arithmetic (Assignment, Addition or Subtraction on Integer, Subtraction of Pointers, Comparing Pointers), Pointers to Pointers, Array of Pointers, Pointers to an Array, Two-dimensional Arrays and Pointers (Passing Two-dimensional Array to a Function), Three-dimensional Arrays, Pointers to Functions (Declaration of a Pointer to a Function, Initialization of Function Pointers, Calling a Function using a Function Pointer, Passing a Function to Another Function, How to Return a Function Pointer, Arrays of Function Pointers).	**25%**
2	**Dynamic Memory Allocation & Advanced Pointer Programming** Dynamic Memory Allocation (Dynamic Allocation of Arrays, Freeing Memory, Reallocating Memory Blocks, Implementing Multidimensional Arrays using Pointers), Offsetting a Pointer, Memory Leak and Memory Corruption, Pointer and Const Qualifier (Pointer to Constant, Constant Pointers, Constant Parameters)	**15%**
3	**User-defined Data Types and Variables: Structures, Unions, Enumerations, Bit-fields.** Structures (Declaring Structures and Structure Variables, Accessing the Members of a Structure, Initialization of Structures, Copying and Comparing Structures, typedef and its Use in Structure Declarations), Nesting of Structures, Arrays of Structures, Initializing Arrays of Structures, Arrays within the Structure, Structures and Pointers, Structures and Functions, Union (Declaring a Union and its Members, Accessing and Initializing Members of a Union), Structure Versus Union, Enumeration Types, Bitfields	**15%**

4	**Files** Files in C (Using Files in C, Declaration of a File Pointer, Opening a File, Closing and Flushing Files) Working with Text Files (Character Input and Output, End of File (EOF), Detecting the End of a File using feof() Function), Working with Binary Files, Direct File Input and Output (Sequential Versus Random File Access), Files of Records (Working with Files of Records) Random Access to Files of Records, Other File Management Functions (Deleting a File, Renaming a File) Low-Level I/O	**15%**
5	**Linked Lists** Singly Linked Lists (Insertion of a Node in a Singly Linked List, Deletion of a Node from a Singly Linked List, Sorting a Singly Linked List, Destroying a Singly Linked List, More Complex Operations on Singly Linked Lists), Circular Linked Lists (Appending a Node, Displaying a Circular Linked List, Inserting a Node after a Specified Node, Inserting a Node before a Particular Node, Deleting a Node, Sorting a Circular Linked List), Doubly Linked Lists (Operations on Doubly Linked Lists, Advantages / Disadvantages of Doubly Linked) Lists, Introduction to Circular Doubly Linked Lists, Applications of Linked Lists (Dynamic Storage Management, Garbage Collection and Compaction), Disadvantages of Linked Lists, Array versus Linked List Revisited	**20%**
6	**Bitwise Operators & Pre-Processors** Bitwise Operator (Bitwise AND, Bitwise OR, Bitwise Exclusive-OR, Bitwise NOT, Bitwise Shift Operator),Command-line Arguments, The C Preprocessor (The C Preprocessor Directives, Predefined Identifiers), Type Qualifier (const Qualifier, volatile Qualifier, restrict Qualifier) Variable Length Argument List, Memory Models and Pointers	**10%**

4. Text Book(s):

1. Pradip Dey, Manas Ghosh, Programming in C, OXFORD,2nd Edition

5. Other Reference Books:

1. Brian Kernighan & Dennis Ritchie, "The C Programming Language", Pearson (2015)
2. Balagurusamy, Programming in ANSI C, Tata McGrawHill, 7Th Edition.
3. Ashok N Kamthane, Programming with ANSI and Turbo C, Pearson
4. Venugopal & Prasad, Mastering C, Tata McGrawHill.
5. Herbert Schildt ,C: The Complete Reference, Tata McGrawHill.
6. Yashwant Kanitkar, Let us C, BPB Publication

6. Chapter wise coverage from textbook

Unit No.	Chapter No. from textbook
1	6.4, 6.10
	07 (7.1 to 7.16)
2	07 (7.17 to 7.20)
3	08
4	09
5	10
6	11

7. Accomplishment

After completion of the course students should become reasonably good at problem solving and algorithm development. They would become capable of solving problems using computers through C programming language.

Practical List

1. Write a C program to sum of Natural Numbers Using Recursion.
2. Write a C program to Display Fibonacci series using Recursion.
3. Write a program in C to demonstrate the use of & (address of) and *(value at address) operator.
4. Write a program in C to add two numbers using pointers.
5. Write a program in C to add numbers using call by reference.
6. Write a program in C to find the maximum number between two numbers using a pointer.
7. Write a program in C to store n elements in an array and print the elements using pointer.
8. Write a program in C to print all permutations of a given string using pointers.
9. Write a program in C to find the largest element using Dynamic Memory Allocation.
10. Write a program in C to calculate the length of the string using a pointer.
11. Write a program in C to swap elements using call by reference.
12. Write a program in C to find the factorial of a given number using pointers
13. Write a program in C to count the number of vowels and consonants in a string using a pointer.
14. Write a program in C to sort an array using Pointer.
15. Write a program in C to compute the sum of all elements in an array using pointers.
16. Write a program in C to print all the alphabets using a pointer.
17. Write a program in C to print a string in reverse using a pointer.
18. Write a C Program to Store Information of a Student Using Structure.
19. Write a C Program to Add Two Distances (in inch-feet) System Using Structures
20. Write a C Program to Calculate Difference Between Two Time Periods
21. Write a C Program to Store Information Using Structures with Dynamically Memory Allocation

22. Create a union that stores an array of 21 characters and 6 ints (6 since 21 / 4 == 5, but 5 * 4 == 20 so you need 1 more for the purpose of this exercise), you will set the integers to 6 given values and then print out the character array both as a series of chars and as a string.
23. Write a program in C to create and store information in a text file.
24. Write a program in C to read an existing file.
25. Write a program in C to write multiple lines in a text file.
26. Write a program in C to read the file and store the lines into an array.
27. Write a program in C to Find the Number of Lines in a Text File.
28. Write a program in C to find the content of the file and number of lines in a Text File.
29. Write a program in C to count a number of words and characters in a file.
30. Write a program in C to delete a specific line from a file.
31. Write a program in C to replace a specific line with another text in a file.
32. Write a program in C to copy a file in another name.
33. Write a program in C to merge two files and write it in a new file.
34. Write a program in C to encrypt a text file and decrypt the encrypted file.
35. Write a program in C to create a singly linked list of n nodes and display it in reverse order.
36. Write a program in C to create a singly linked list of n nodes and count the number of nodes.
37. Write a program in C to insert a new node at the beginning of a Singly Linked List.
38. Write a program in C to insert a new node at the end of a Singly Linked List.
39. Write a program in C to insert a new node at the middle of Singly Linked List.
40. Write a program in C to delete first node of Singly Linked List.
41. Write a program in C to delete a node from the middle of Singly Linked List.
42. Write a program in C to delete the last node of Singly Linked List.
43. Write a program in C to search an existing element in a singly linked list.
44. Write a program in C to create a doubly linked list and display in reverse order.
45. Write a program in C to insert a new node at the end of a doubly linked list.
46. Write a program in C to insert a new node at any position in a doubly linked list.
47. Write a program in C to delete a node from any position of a doubly linked list.
48. Write a program in C to find the maximum value from a doubly linked list.
49. Write a program in C to insert a node at the end of a circular linked list.
50. Write a program in C to search an element in a circular linked list.
51. Write a C program to get the rightmost bit of any input.
52. Write a C program to remove leftmost bit of any input
53. Write a C program to remove first bit of any input, and add it to the right.
54. Write a C program to rotate bits of a given number.
55. Write a C program to convert decimal to binary number system using bitwise operator.
56. Write a C program to swap two numbers using bitwise operator.
57. Write a C program to check whether a number is even or odd using bitwise operator.
58. Give an array of positive integers, find the total sum after performing the bit wise OR operation on all the sub arrays of a given array.