GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Master of Computer Applications, 1st Semester
Subject Name: Problem Solving using C
Subject Code: 619401

With effective from academic year 2020-21

1. **Learning Objectives:**

   The student will be able to develop solution of problem using his/her own logic and gradually learn to solve various problems using efficient algorithm and Programming techniques.

2. **Prerequisites:** Logical Thinking Ability, School-level mathematics

3. **Mode of Delivery**:
   In tutorial sessions, discuss the basics and the methodology for tasks to be completed during next lab session(s). In the Lab sessions – Ask students to write, compile and test the programs on computer. Check for common difficulties faced by students. Conduct a brief session to explain and then resume the hands-on working. If a few students are not able to complete all the assigned task(s) during lab sessions, help them and motivate them to complete pending tasks at home.

4. **Course Contents:**

Tasks #	***Topics (Programs) to be Completed***	***Est. Hrs.***
1.	* Introduction: Objectives; Methodology; C Language Constructs; Good Programming Practices Write, compile and execute the following programs: * Simple programs-1: Print “Hello world!”; Print “My name is <Name>”; Print “<Address Lines>” Explain the concept of variables, their declaration, and Arithmetic Assignment statements * Simple programs-2a: x1 = 2; x2 = 7; Print “x1 = value, y1 = value”. * Write a program to take 5 integers and find and print the total and average of the 5 numbers. Repeat the same for floating point numbers instead of integers. Explain for-loop. Also explain the general algorithm for summation: (i) Initialize sum = 0; (ii) repeat sum = sum + value * Write a program to find the sum of 1, 2, 3, … , n. Print average (avg) also. Explain scanf() for accepting user inputs. * Write a program to accept n. Find sum of n values accepted 1-by-1. Also find average (avg). Print sum, avg. Additionally, print the input values also. Explain Arrays to store multiple values * Write a program to accept n and n input values to be stored in an array. Find sum and average (avg) of n values. Print input values followed by sum, avg.	2 + 4
2.	Explain string variable, its declaration and scanf() to accept strings	2 + 4

	* Write a program to accept as input: first name, middle name, surname; then print name, first as (a) first mid surname; and thereafter (b) surname first mid, etc in Explain algorithm to find string length. Explain: if-statements: if ( ) {… }; if ( ) {… } else {… }; and if ( ) {… } else if ( ) {… }; * Write a program to find string length. Is the string length same as number of characters in the string? Explain also the concept of function. * Rewrite the program to find string length by using function for finding string length. Test this program to find lengths of first, mid, and surname. * Write a program to print a given string in reverse order.	
3.	Explain: Convert temperature in degrees Fahrenheit to degrees Celsius and vice versa. * Write a program to take Input: n, and n values of temperature in °F, convert these into °C and print the values in a table with 1st column containing °F and 2nd column °C Explain the concept of function() in C. * Modify the above program to convert temperature in °C into °F. Write a function c2f() for this operation. How many input arguments are there? How many outputs from the function?	1 + 2
4.	Explain the method of checking of “Divisibility of an integer by another integer” – Modulus operator (%). For segregating a given list of integers into even and odd numbers using bitwise operations. * Write a program in C: Inputs: 2 integer values: Numerator (num1) and Denominator (num2). Output: Quotient (q) and Remainder (r). * Write a program in C: Input: An array (a List) of n integers. Output: * Develop an algorithm write a program in C to print all ***primes*** in the first n (n > 1) integers. Develop the most efficient algorithm.	2 + 4
5.	Explain Euler’s algorithm to find gcd(m, n). * Given two integers m and n (m, n >= 0), develop an algorithm and write a program in C to find their greatest common divisor (gcd) * Write a program in C to rearrange the elements in an array so that they appear in reverse order.	2 + 4 Cum. 9+18
6.	* Write a program to calculate and display the value of the slope of the line connecting the two points whose coordinates are (3, 7) and (8, 12). Slope of a line between two points (x_1 , y_1) and (x_2 , y_2) is (y_2 – y_1) / (x_2 – x_l). Run the same program for the line connecting the points (2, 10) and (12, 6), and other pairs of points.	1 + 2

	* Write a program to calculate and display the coordinates of the midpoint of the line connecting the two points given in the previous Exercise. The coordinates of the midpoint between two points having coordinates (x_1 , y_1) and (x_2 , y_2) are ((x_1 + x_2) / 2, (y_1 + y_2) / 2). * Write a program that calculates the distance between two points whose coordinates are (7, 12) and (3, 9). Distance between two points having coordinates (x_1 , y_1) and (x_2 , y_2) = sqrt([x_1 – x_2]^2 + [y_1 – y_2]^2). Also, run the program for the points (-12, -15) and (22, 5) and a few other points.	
7.	Introduce O(n) notation as an indicator of how fast a program works. * Given some integer x, develop an algorithm and write a program to compute the value of x^n where n is considerably larger than 1. This algorithm has time complexity O(n). * Develop an improved algorithm having time complexity O($\log_2$n).	2 + 4
8.	Explain: (a) sin(x) defined by the infinite series expansion. What will be the stopping criterion? * Write a program to evaluate sin(x) * Write a program to determine and display the maximum height reached when the ball is thrown at 5 miles / hour at an angle of 60 degrees. (Hint: Make sure to convert the initial velocity into the correct units.) The maximum height reached by a ball thrown with an initial velocity v in feet/sec at an angle of θ is given by the formula height = (0.5 * v^2 * $\sin^2 \theta$) / 32.2. Run the program for v = 7 miles / hour and angle = 45 degrees. Explain the problems and the solution approach. * Write a program to calculate and print the height (h) = L * sin θ , where L is the Length of the Ladder, and θ is the angle the ladder makes with the horizontal base. Data: (a) L = 20 feet, θ = 85^o , (b) L = 25 feet, θ = 75^o . * Write a program that calculates the x and y coordinates of the point whose polar coordinates are r = 10 and θ = 30^o , using the following formulas: x = r cos θ and y = r sin θ . Run the program again to convert polar coordinates: r = 12.5 and θ = 67.8° into rectangular coordinates.	2 + 4
9.	* Given an integer n >= 1, develop an algorithm and write a program to find the smallest exact divisor of n other than one. * Every integer can be expressed as a product prime numbers. Develop an algorithm and write a program to compute all the prime factors of a given integer n > 0.	1 + 2 Cum. 15+30

10.	* Write a program in C (a) to find factorial of n (n!), and (b) first n terms of the Fibonacci sequence using an iterative algorithm. Explain Recursion formulation of a problem and its working by taking 3 examples: (a) n! ; (b) fibo(n); and (c) sum of $a_0 + a_1 + \ldots + a_{n-1} + a_n$. * Write a program in C (a) to find factorial of n (n!); (b) first n terms of the Fibonacci sequence using an iterative algorithm; and (c) sum of $a_0 + a_1 + \ldots + a_{n-1} + a_n$.	2 + 4
11.	Explain: Exchange the values of two variables. * Program : Input: a, b Output: a (prints the value of b), b (prints the value of a) Explain preliminary concept of pointers and write a few simple programs to help understanding the concept of ***pointers***: * Implement the program in C for “exchanging the values of two variables” using function (which will require use of pointers for function arguments in C) * Write a C program to find sum of n values a_i , i = 1 to n, using pointers instead of arrays. * Write a C program to count number of words in a given text by representing text string as pointer instead of array.	3 + 6
12.	* Remove all duplicates from an ordered array and contract the array accordingly.	1 + 2
13.	Explain: Numerical methods to find sqrt(n). * Given a number n >= 0, develop an algorithm and write a program in C to compute square root of a given non-negative number (n >= 0) by Divide-and-Conquer method * Write a program in C using an improved algorithm to compute Square Root using Newton’s method and other methods.	2 + 4
14.	* Write a C program to find Maximum and Minimum values in a given array (or List) of values. Also write the C program using pointers instead of array.	1 + 2
	Total Hours	24 + 48

5. **Text Book**:
    1. Brian W. Kernigham, Dennis Ritchie, “The C Programming Language”, Pearson (2015)
    2. R. G. Dromey, “How to Solve it by Computer”, Pearson (2013)

6. **Reference Books**:
   - Yeshvant Kanetkar, “Let Us C”, BPB Publication (2017)

   Webliography

   - https://www.w3resource.com/c-programming-exercises/
   - https://www.codechef.com/
   - https://www.learn-c.org/
   - https://www.prep.youth4work.com/practice-tests/c-programming-test
   - https://www.indiabix.com/online-test/c-programming-test/
   - https://www.hackerrank.com/c-programming-test-1/
   - https://www.mycplus.com/featured-articles/programming-contests-and-challenges/
   - https://www.hackerearth.com/challenges/
   - https://www.geeksforgeeks.org/category/competitive-programming/
   - https://www.techgig.com/challenge
   - https://www.freecodecamp.org/news/the-10-most-popular-coding-challenge-websites-of-2016-fb8a5672d22f/

7. **Course Outcome**:
   After studying this course, students will be able to write programs using iterative and recursive algorithms for various basic tasks such as to carry out the following tasks:
   - Sum and average of a given sequence of numbers using an array (a Pointer) or a List
   - Sum of infinite series, such as for trigonometric functions, etc
   - Using numerical methods to find Square root of a number
   - Prime numbers, prime divisors of n
   - GCD of given integers
   - Find n! and first n Fibonacci numbers using iterative and recursive algorithms
   - Find maximum and minimum in a given sequence on n numbers.
   - Remove duplicate values in an array.
   - Using Pointers, exchange 2 values
   - Improved algorithm for x^n

8. **Active Learning Assignment**
   - Simulate a simple dictionary. Assume each character contains at least 10 vocabularies. Create an index page for all characters. Retrieve the word using index value. Assume that the index characters are from a to z.
   - Design a simple search engine to display the possible websites upon entering a search query. Use suitable data structure for storage and retrieval

9. **Additional Exercises:**
   1. Given two variables of integer type a and b, exchange their values without using a third temporary variable.
   2. Write a program to print out n values of the following sequence: 1 –1 1 –1 1 –1 …
   3. Write a program to compute the sum of the first n terms (n >= 1) of the following series: s = 1 – 3 + 5 – 7 + 9 – 11 + …
   4. For a given x and n, design an algorithm to compute x^n / n! .

5. Design an algorithm to compute ${}^{n}C_{r}$ = n! / [r! (n – r)!] for a given value of n and r.
    a. The exponential growth constant e is characterized by the infinite series:
       e = 1 / 0! + 1 / 1! + 1 / 2! + 1 / 3! + 1 / 4! + …
6. Design an algorithm to check whether two given integers are consecutive numbers of Fibonacci sequence or not.
7. Design an algorithm that computes (a) counts of number of digits, and (b) sum of digits in an integer.
8. Design an algorithm that reads in a set of n single digits and convert them into a single decimal integer. For example, the algorithm should convert a sequence of 6 digits (3, 2, 6, 4, 8, 5) to the integer 326485.
9. Design an algorithm that converts a given binary number to (a) Octal number and (b) Hexadecimal number.
10. Design an algorithm that accepts as input a decimal fraction and converts it into a corresponding binary fraction of a fixed accuracy. (For example 0.625_{10} = 0.101_{2} = 1 x 2^{-1} + 0 x 2^{-2} + 1 x 2^{-3})
11. Given that all ASCII codes are less than 128, design an algorithm that reads a given set of data and decides whether or not it may contain decimal data.
12. Design an algorithm that finds an integer whose square is closest to but greater than the integer number input as data.
13. For the integers in the range 1 to n (take n = 100), find the number that has the most divisors.
14. It is well known that adjacent Fibonacci numbers do not share a common divisor greater than 1 (they are relatively prime). Design an algorithm that tests this observation for the first n integers.
15. A perfect number is one whose divisors add up to the number. Design and implement an algorithm that determines all perfect numbers between 1 and 500. (Ex: 6 = 1 x 2 x 3; 6 = 1 + 2 + 3).
16. It is possible to compute n! In O (log2n) steps. Develop such an algorithm for computing n!.
17. Design an algorithm that rearranges the elements of an array such that all those numbers originally stored at odd suffixes are placed before those stored at even suffixes. For example, the input array 1, 2, 3, 4, 5, 6, 7, 8 will be transformed to 1, 3, 5, 7, 2, 4, 6, 8.
18. Generate Following Pattern.

1
```
N=  4
1
1   2
1   2   3
1   2   3   4
```

2.
```
N=  4
A
A   B
A   B   C
A   B   C   D
```

3.
```
      N=  4
                A
            A   B   A
        A   B   C   B   A
    A   B   C   D   C   B   A
```

4.
```
n=  4
                0
            1   0   1
        2   1   0   1   2
    3   2   1   0   1   2   3
```

5.

```
N=  4
            *
        *   *   *
    *   *   *   *   *
*   *   *   *   *   *   *
    *   *   *   *   *
        *   *   *
            *
```

6.

```
N=  4
            1
        1   2   1
    1   2   3   2   1
1   2   3   4   3   2   1
    1   2   3   2   1
        1   2   1
            1
```

7.

```
N=  5
                +
            -   -
        *   *   *
    +   +   +   +
-   -   -   -   -
```

8.

```
N=  5
*                                *
*   *                        *   *
*   *   *                *   *   *
*   *   *   *        *   *   *   *
*   *   *   *   *    *   *   *   *
```

9.

```
n=  4
            *
        *       *
    *       *       *
*       *       *       *
    *       *       *
        *       *
            *
```