EST 102	PROGRAMING IN C	CATEGORY	L	T	P	CREDIT	YEAR OF INTRODUCTION
		ESC	2	1	2	4	2019

Preamble: The syllabus is prepared with the view of preparing the Engineering Graduates capable of writing readable C programs to solve computational problems that they may have to solve in their professional life. The course content is decided to cover the essential programming fundamentals which can be taught within the given slots in the curriculum. This course has got 2 Hours per week for practicing programming in C. A list showing 24 mandatory programming problems are given at the end. The instructor is supposed to give homework/assignments to write the listed programs in the rough record as and when the required theory part is covered in the class. The students are expected to come prepared with the required program written in the rough record for the lab classes.

Prerequisite: NIL

Course Outcomes: After the completion of the course the student will be able to

CO 1	Analyze a computational problem and develop an algorithm/flowchart to find its solution
CO 2	Develop readable* C programs with branching and looping statements, which uses Arithmetic, Logical, Relational or Bitwise operators.
CO 3	Write readable C programs with arrays, structure or union for storing the the data to be processed
CO 4	Divide a given computational problem into a number of modules and develop a readable multi-function C program by using recursion if required, to find the solution to the computational problem
CO 5	Write readable C programs which use pointers for array processing and parameter passing
CO 6	Develop readable C programs with files for reading input and storing output

readable* - readability of a program means the following:
1. Logic used is easy to follow
2. Standards to be followed for indentation and formatting
3. Meaningful names are given to variables
4. Concise comments are provided wherever needed

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	✓	✓	✓	✓		✓				✓	✓	✓
CO2	✓	✓	✓	✓	✓					✓		✓
CO3	✓	✓	✓	✓	✓					✓		✓
CO4	✓	✓	✓	✓	✓					✓	✓	✓
CO5	✓	✓			✓					✓		✓
CO6	✓	✓			✓					✓		✓

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination Marks
	Test 1 (Marks)	Test 2 (Marks)	
Remember	15	10	25
Understand	10	15	25
Apply	20	20	40
Analyse	5	5	10
Evaluate			
Create			

Mark distribution

Total Marks	CIE Marks	ESE Marks	ESE Duration
150	50	100	3 hours

Continuous Internal Evaluation Pattern:

Attendance   : 10 marks

Continuous Assessment Test 1 (for theory, for 2 hrs)   : 20 marks

Continuous Assessment Test 2 (for lab, internal examination, for 2 hrs)   : 20 marks

Internal Examination Pattern: There will be two parts; Part A and Part B. Part A contains 5 questions with 2 questions from each module (2.5 modules x 2 = 5), having 3 marks for each question. Students should answer all questions. Part B also contains 5 questions with 2 questions from each module (2.5 modules x 2 = 5), of which a student should answer any one. A question should not have sub-divisions and each one carries 7 marks.

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 questions with 2 questions from each module, having 3 marks for each question. Students should answer all questions. Part B contains 2 questions from each module of which a student should answer any one. Each question can have maximum 2 sub-divisions and carry 14 marks.

Sample Course Level Assessment Questions

Course Outcome 1 (CO1): Write an algorithm to check whether largest of 3 natural numbers is prime or not. Also, draw a flowchart for solving the same problem.

Course Outcome 2 (CO2): Write an easy to read C program to process a set of n natural numbers and to find the largest even number and smallest odd number from the given set of numbers. The program should not use division and modulus operators.

Course Outcome 3(CO3): Write an easy to read C program to process the marks obtained by n students of a class and prepare their rank list based on the sum of the marks obtained. There are 3 subjects for which examinations are conducted and the third subject is an elective where a student is allowed to take any one of the two courses offered.

Course Outcome 4 (CO4): Write an easy to read C program to find the value of a mathematical function f which is defined as follows. f(n) = n! / (sum of factors of n), if n is not prime and f(n) = n! / (sum of digits of n), if n is prime.

Course Outcome 5 (CO5): Write an easy to read C program to sort a set of n integers and to find the number of unique numbers and the number of repeated numbers in the given set of numbers. Use a function which takes an integer array of n elements, sorts the array using the Bubble Sorting Technique and returns the number of unique numbers and the number of repeated numbers in the given array.

Course Outcome 6 (CO6): Write an easy to read C program to process a text file and to print the Palindrome words into an output file.

Model Question paper

QP CODE: PAGES:3

Reg No:_____

Name :_____

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY FIRST SEMESTER B.TECH DEGREE EXAMINATION, MONTH & YEAR

Course Code: **EST 102**

Course Name: Programming in C (Common to all programs)

Max.Marks:100   **Duration: 3 Hours**

PART A

Answer all Questions. Each question carries 3 Marks

Part A

(*Answer all questions. Each question carries 3 marks*)

1.   Write short note on processor and memory in a computer.

2.   What are the differences between compiled and interpreted languages? Give example for each.

3.   Write a C program to read a Natural Number through keyboard and to display the reverse of the given number. For example, if "3214567" is given as input, the output to be shown is "7654123".

4.   Is it advisable to use *goto* statements in a C program? Justify your answer.

5.   Explain the different ways in which you can *declare & initialize* a single dimensional array.

6.   Write a C program to read a sentence through keyboard and to display the count of white spaces in the given sentence.

7.   What are the advantages of using functions in a program?

8.   With a simple example program, explain *scope* and *life time* of variables in C.

9.   Write a function in C which takes the address of a single dimensional array (containing a finite sequence of numbers) and the number of numbers stored in the array as arguments and stores the numbers in the same array in reverse order. Use pointers to access the elements of the array.

10. With an example, explain the different modes of opening a file.   (10x3=30)

Part B

11.

(a) Draw a flow chart to find the position of an element in a given sequence, using linear searching technique. With an example explain how the flowchart finds the position of a given element. **(10)**

(b) Write a pseudo code representing the flowchart for linear searching. **(4)**

OR

12.

(a) With the help of a flow chart, explain the bubble sort operation. Illustrate with an example. **(10)**

(b) Write an algorithm representing the flowchart for bubble sort. **(4)**

13.

(a) Write a C program to read an English Alphabet through keyboard and display whether the given Alphabet is in upper case or lower case. **(6)**

(b) Explain how one can use the builtin function in C, *scanf* to read values of different data types. Also explain using examples how one can use the builtin function in C, *printf* for text formatting. **(8)**

OR

14.

(a) With suitable examples, explain various operators in C. **(10)**

(b) Explain how characters are stored and processed in C. **(4)**

15.

(a) Write a function in C which takes a 2-Dimensional array storing a matrix of numbers and the order of the matrix (number of rows and columns) as arguments and displays the sum of the elements stored in each row. **(6)**

(b) Write a C program to check whether a given matrix is a diagonal matrix. **(8)**

OR

16.

(a) Without using any builtin string processing function like *strlen*, *strcat* etc., write a program to concatenate two strings. **(8)**

(b) Write a C program to perform bubble sort. **(6)**

17.

    (a) Write a function namely *myFact* in C to find the factorial of a given number. Also, write another function in C namely*nCr* which accepts two positive integer parameters *n* and *r* and returns the value of the mathematical function*C(n,r)*(n! / (r! x (n - r)!)). The function *nCr* is expected to make use of the factorial function *myFact.* **(10)**

    (b) What is recursion? Give an example. **(4)**

<div align="center">OR</div>

18.

    (a) With a suitable example, explain the differences between a structure and a union in C. **(6)**

    (b) Declare a structure namely *Student* to store the details (*roll number, name, mark_for_C*) of a student. Then, write a program in C to find the average mark obtained by the students in a class for the subject *Programming in C* (using the field *mark_for_C*). Use array of structures to store the required data **(8)**

19.

    (a) With a suitable example, explain the concept of pass by reference. **(6)**

    (b) With a suitable example, explain how pointers can help in changing the content of a single dimensionally array passed as an argument to a function in C. **(8)**

<div align="center">OR</div>

20.

    (a) Differentiate between sequential files and random access files? **(4)**

    (b) Using the prototypes explain the functionality provided by the following functions. **(10)**

        i.   *rewind()*

        ii. *fseek()*

        iii. *ftell()*

        iv. *fread()*

        v. ***fwrite()*** *(14X5=70)*

SYLLABUS
Programming in C (Common to all disciplines)

Module 1
Basics of Computer Hardware and Software

Basics of Computer Architecture: processor, Memory, Input& Output devices

Application Software & System software: Compilers, interpreters, High level and low level languages

Introduction to structured approach to programming, Flow chart Algorithms, Pseudo code (*bubble sort, linear search - algorithms and pseudocode*)

Module 2
Program Basics

Basic structure of C program: Character set, Tokens, Identifiers in C, Variables and Data Types , Constants, Console IO Operations, printf and scanf

Operators and Expressions: Expressions and Arithmetic Operators, Relational and Logical Operators, Conditional operator, size of operator, Assignment operators and Bitwise Operators. Operators Precedence

Control Flow Statements: If Statement, Switch Statement, Unconditional Branching using goto statement, While Loop, Do While Loop, For Loop, Break and Continue statements.(Simple programs covering control flow)

Module 3
Arrays and strings

Arrays Declaration and Initialization, 1-Dimensional Array, 2-Dimensional Array

String processing: In built String handling functions (strlen, strcpy, strcat and strcmp, puts, gets)

Linear search program, bubble sort program, simple programs covering arrays and strings

Module 4
Working with functions

Introduction to modular programming, writing functions, formal parameters, actual parameters

Pass by Value, Recursion, Arrays as Function Parameters structure, union, Storage Classes, Scope and life time of variables, *simple programs using functions*

Module 5
Pointers and Files

Basics of Pointer: declaring pointers, accessing data though pointers, NULL pointer,array access using pointers, pass by reference effect

File Operations: open, close, read, write, append

Sequential access and random access to files: In built file handlingfunctions (*rewind() ,fseek(), ftell(), feof(), fread(), fwrite()*), *simple programs covering pointers and files.*

Text Books

1. Schaum Series, Gottfried B.S.,Tata McGraw Hill,Programming with C
2. E. Balagurusamy, Mcgraw Hill,Programming in ANSI C
3. Asok N Kamthane, Pearson,Programming in C
4. Anita Goel, Pearson, Computer Fundamentals

Reference Books

1. Anita Goel and Ajay Mittal, Pearson, Computer fundamentals and Programming in C
2. Brian W. Kernighan and Dennis M. Ritchie, Pearson, C Programming Language
3. Rajaraman V, PHI, Computer Basics and Programming in C
4. Yashavant P, Kanetkar, BPB Publications, Let us C

Course Contents and Lecture Schedule

Module 1: Basics of Computer Hardware and Software		(7 hours)
1.1	**Basics of Computer Architecture:**Processor, Memory, Input& Output devices	2 hours
1.2	**Application Software & System software:**Compilers, interpreters,High level and low level languages	2 hours
1.3	Introduction to structured approach to programming, Flow chart	1 hours
1.4	Algorithms, Pseudo code (*bubble sort, linear search - algorithms and pseudocode*)	2 hours
Module 2: Program Basics		(8 hours)
2.1	**Basic structure of C program:** Character set,Tokens, Identifiers in C, Variables and Data Types , Constants, Console IO Operations,printf and scanf	2 hours
2.2	**Operators and Expressions:**Expressions and Arithmetic Operators, Relational and Logical Operators,Conditional operator, sizeof operator, Assignment operators and Bitwise Operators. Operators Precedence	2 hours
2.3	**Control Flow Statements:** If Statement, Switch Statement, Unconditional Branching using goto statement, While Loop,Do While Loop,For Loop, Break and Continue statements.*(Simple programs covering control flow)*	4 hours
Module 3: Arrays and strings:		(6 hours)
3.1	Arrays Declaration and Initialization, 1-Dimensional Array, 2-Dimensional Array	2 hours

3.2	**String processing:** In built String handlingfunctions(*strlen, strcpy, strcat and strcmp, puts, gets*)	**2 hours**
3.3	Linear search program, bubble sort program,*simple programs covering arrays and strings*	**3 hours**
Module 4: Working with functions		**(7 hours)**
4.1	Introduction to modular programming, writing functions, formal parameters, actual parameters	**2 hours**
4.2	Pass by Value, Recursion, Arrays as Function Parameters	**2 hours**
4.3	structure, union, Storage Classes,Scope and life time of variables, *simple programs using functions*	**3 hours**
Module 5: Pointers and Files		**(7 hours)**
5.1	**Basics of Pointer**: declaring pointers, accessing data though pointers, NULL pointer,array access using pointers, pass by reference effect	**3 hours**
5.2	**File Operations**: open, close, read, write, append	1 **hours**
5.3	**Sequential access and random access to files:** In built file handlingfunctions (*rewind() ,fseek(), ftell(), feof(), fread(), fwrite()*), *simple programs covering pointers and files.*	**2 hours**

C PROGRAMMING LAB (Practical part of EST 102, Programming in C)

Assessment Method: The Academic Assessment for the Programming lab should be done internally by the College. The assessment shall be made on 50 marks and the mark is divided as follows: Practical Records/Outputs - 20 marks (internal by the College), Regular Lab Viva - 5 marks (internal by the College), Final Practical Exam – 25 marks (internal by the College).

LIST OF LAB EXPERIMENTS
1.Familiarization of Hardware Components of a Computer
2. Familiarization of Linux environment – How to do Programming in C with Linux
3.Familiarization of console I/O and operators in C
　　　　i) Display "Hello World"
　　　　ii) Read two numbers, add them and display theirsum
　　　　iii) Read the radius of a circle, calculate its area and display it

iv)Evaluate the arithmetic expression ((a -b / c * d + e) * (f +g)) and display its solution.Read the values of the variables from the user through console.

1. Read 3 integer values and find the largest amoung them.
2.Read a Natural Number and check whether the number is prime or not
3.Read a Natural Number and check whether the number is Armstrong or not
4.Read n integers, store them in an array and find their sum and average
5.Read n integers, store them in an array and search for an element in the array using an algorithm for Linear Search
6.Read n integers, store them in an array and sort the elements in the array using Bubble Sort algorithm
7.Read a string (word), store it in an array and check whether it is a palindrome word or not.
8.Read two strings (each one ending with a $ symbol), store them in arrays and concatenate them without using library functions.
9.Read a string (ending with a $ symbol), store it in an array and count the number of vowels, consonants and spaces in it.
10.Read two input each representing the distances between two points in the Euclidean space, store these in structure variables and add the two distance values.
11.Using structure, read and print data of n employees (*Name, Employee Id and Salary*)
12.Declare a union containing 5 string variables (*Name, House Name, City Name, State and Pin code*) each with a length of C_SIZE (user defined constant). Then, read and display the address of a person using a variable of the union.
13.Find the factorial of a given Natural Number n usingrecursive and non recursive functions
14.Read a string (word), store it in an array and obtain its reverse by using a user defined function.
15.Write a menu driven program for performing matrix addition, multiplication and finding the transpose. Use functions to (i) read a matrix, (ii) find the sum of two matrices, (iii) find the product of two matrices, (i) find the transpose of a matrix and (v) display a matrix.
16.Do the following using pointers

      i) add two numbers

      ii) swap two numbers using a user defined function

17.Input and Print the elements of an array using pointers

18.Compute sum of the elements stored in an array using pointers and user defined function.

19.Create a file and perform the following

      iii) Write data to the file

      iv) Read the data in a given file & display the file content on console

      v) append new data and display on console

20.Open a text input file and count number of characters, words and lines in it; and store the results in an output file.