

**RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)**

B.TECH. DEGREE PROGRAMME

**SECOND SEMESTER
(2020 ADMISSIONS)**

100908/CO200G	PROGRAMMING IN C
----------------------	-------------------------

SYLLABUS

Rajagiri Valley, Kakkanad,
Kochi 682 039, Kerala, INDIA
www.rajagiritech.ac.in

COURSE CODE	COURSE NAME	L	T	P	CREDIT	YEAR OF INTRODUCTION
100908/CO200G	PROGRAMMING IN C	2	1	2	4	2020

1. **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.

2. **Prerequisite:** NIL

3. **Syllabus:**

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, Pseudocode (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 through 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 handling functions (rewind(), fseek(), ftell(), feof(), fread(), fwrite()), simple programs covering pointers and files.

4. 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

5. 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
6. **Course Outcomes:** After the completion of the course the student will be able to

CO1: Analyze a computational problem and develop an algorithm/flowchart to find its solution

CO2: Develop readable* C programs with branching and looping statements, which uses Arithmetic, Logical, Relational or Bitwise operators.

CO3: Write readable C programs with arrays, structure or union for storing the data to be processed

CO4: 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

CO5: Write readable C programs which use pointers for array processing and parameter passing

CO6: 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

7. **Mapping of course outcomes with program outcomes:**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	2		1	-	-	-	1	1	2
CO2	2	2	2	1	1	-	-	-	-	1	-	2
CO3	2	2	2	1	2	-	-	-	-	1	-	2
CO4	3	3	3	2	3	-	-	-	-	1	1	2
CO5	3	2	-	-	2	-	-	-	-	1	-	2
CO6	3	3	-	-	3					3		3

8. **Assessment Pattern:**

Learning Objectives	Continuous Internal Evaluation (CIE)		End Semester Examination (ESE out of 100)
	Internal Examination 1 (25)	Internal Examination 2 (25)	

Remember	7.5	5	25
Understand	5	7.5	25
Apply	10	10	40
Analyze	2.5	2.5	10
Evaluate			

9. Mark Distribution

Total	CIE				ESE
	Attendance	Continuous Assessment Test 1 (for theory)	Continuous Assessment Test 2 (for lab, internal examination)	Total	
150	10	20 (Average of 2 series tests)	20	50	100

10. 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.

11. C PROGRAMMING LAB (Practical part of 100908/CO200G, Programming in C)

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

The mark obtained out of 50 will be converted into equivalent proportion out of 20 for CIE computation.

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 their sum
- 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.

4. Read 3 integer values and find the largest among them.
5. Read a Natural Number and check whether the number is prime or not
6. Read a Natural Number and check whether the number is Armstrong or not
7. Read n integers, store them in an array and find their sum and average
8. Read n integers, store them in an array and search for an element in the array using an algorithm for Linear Search
9. Read n integers, store them in an array and sort the elements in the array using Bubble Sort algorithm
10. Read a string (word), store it in an array and check whether it is a palindrome word or not.
11. Read two strings (each one ending with a \$ symbol), store them in arrays and concatenate them without using library functions.
12. Read a string (ending with a \$ symbol), store it in an array and count the number of vowels, consonants and spaces in it.
13. 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.
14. Using structure, read and print data of n employees (Name, Employee Id and Salary)
15. 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.
16. Find the factorial of a given Natural Number n using recursive and non-recursive functions
17. Read a string (word), store it in an array and obtain its reverse by using a user defined function.
18. 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, (iv) find the transpose of a matrix and (v) display a matrix.
19. Do the following using pointers
 - i. add two numbers
 - ii. swap two numbers using a user defined function

20. Input and Print the elements of an array using pointers
21. Compute sum of the elements stored in an array using pointers and user defined function.
22. Create a file and perform the following
 - i. Write data to the file
 - ii. Read the data in a given file & display the file content on console
 - iii. append new data and display on console
23. Open a text input file and count number of characters, words and lines in it; and store the results in an output file.