

Program Name : Computer Engineering Program Group
Program Code : CO/CM/IF/CW/IF
Semester : Second
Course Title : Programming in 'C'
Course Code : 22226

1. RATIONALE

Diploma engineers (also called technologists) have to write programs to cater with various IT solutions. In order to develop a program to solve a given problem, they have to build logic, develop algorithms and flow charts. This course is designed keeping in view developing these skills. Besides its use to write codes for low level programming such as developing operating systems, drivers, and compilers; 'C' has been widely used as a general-purpose language to develop basic applications. This course deals with fundamental syntactic information about 'C' that will help the students to apply the basic concepts, program structure and principles of 'C' programming paradigm to build given application. The course is basically designed to create a base to develop foundation skills of programming language.

2. COMPETENCY

The aim of this course is to help the student to attain the following industry identified competency through various teaching learning experiences:

- Develop 'C' programs to solve broad-based computer related problems.

3. COURSE OUTCOMES (COs)

The theory, practical experiences and relevant soft skills associated with this course are to be taught and implemented, so that the student demonstrates the following *industry oriented* COs associated with the above mentioned competency:

- Develop flowchart and algorithm to solve problems logically.
- Write simple 'C' programs using arithmetic expressions.
- Develop 'C' programs using control structure.
- Develop 'C' programs using arrays and structures.
- Develop/Use functions in C programs for modular programming approach.
- Develop 'C' programs using pointers.

4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme			Credit (L+T+P)	Examination Scheme												
L	T	P		Theory					Practical							
				Paper Hrs.	ESE Max	ESE Min	PA Max	PA Min	Total Max	Total Min	ESE Max	ESE Min	PA Max	PA Min	Total Max	Total Min
3	2	2	7	3	70	28	30*	00	100	40	25@	10	25	10	50	20

(*): Under the theory PA, Out of 30 marks, 10 marks are for micro-project assessment to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be taken during the semester for the assessment of the UOs required for the attainment of the COs.
Legends: L-Lecture; T - Tutorial/Teacher Guided Theory Practice; P - Practical; C - Credit
ESE - End Semester Examination; **PA** - Progressive Assessment



5. COURSE MAP (with sample COs, PrOs, UOs, ADOs and topics)

This course map illustrates an overview of the flow and linkages of the topics at various levels of outcomes (details in subsequent sections) to be attained by the student by the end of the course, in all domains of learning in terms of the industry/employer identified competency depicted at the centre of this map.

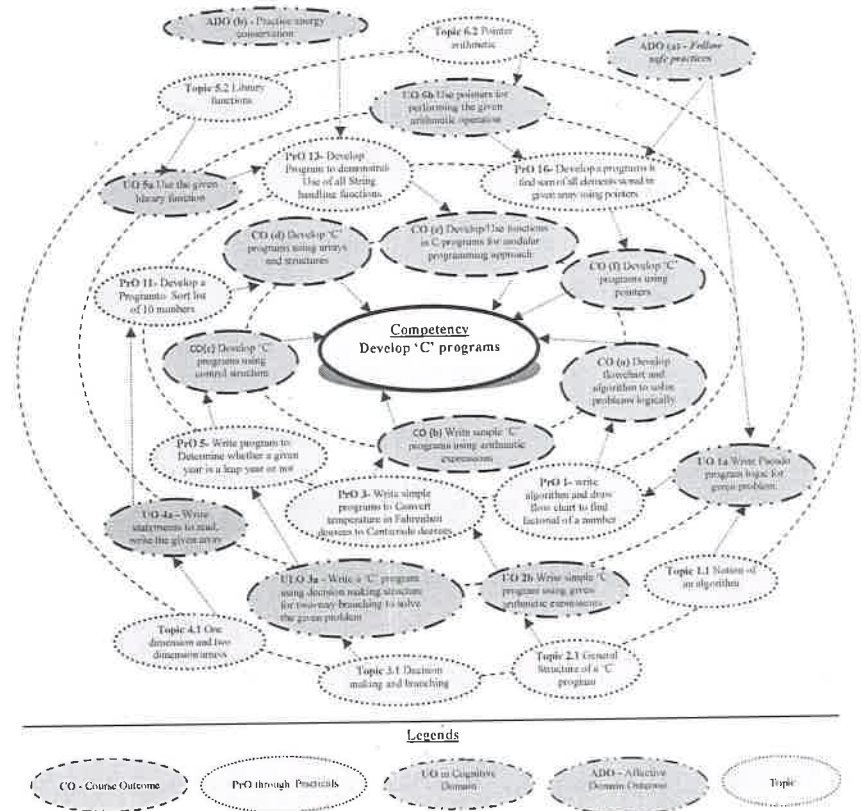


Figure 1 - Course Map

6. SUGGESTED PRACTICALS/ EXERCISES

The practicals in this section are PrOs (i.e. sub-components of the COs) to be developed and assessed in the student for the attainment of the competency.

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
1	Write/compile/execute simple 'C' program: Develop minimum 2 programs using Constants, Variables, arithmetic expression.	II	02
2	Write/compile/execute simple 'C' program: Develop minimum	II	02

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
	2 programs increment/decrement operators, exhibiting data type conversion		
3	Write simple programs to convert temperature in Fahrenheit <u>degrees</u> to Centi <u>grade degrees</u> .	II	2
4	Write simple programs to calculate the area and perimeter of the <u>rectangle</u> , and the area & circumference of the circle	II	2
5	Decision Making and branching using if, if-else structure Write program to: (i) Determine whether a given year is a leap year or not (ii) Determine whether a string is <u>palindrome</u> .	III	02*
6	Write program to: (i) Find the greatest of the three numbers using conditional operators (ii) Find if a <u>given character</u> is vowel.	III	02
7	Using switch statement: Write programs to : (i) Print day of week by taking number from 1 to 7 (ii) Print a student's <u>grade by accepting percent marks</u> .	III	02
8	Using switch statement: Write programs to check whether the <u>triangle is isosceles, equilateral, scalene or right angled triangle</u>	III	02
9	Looping: Write a program to : (i) Find sum of digits of a given number. (ii) Generate <u>multiplication table up to 10 for numbers 1 to 5</u> .	III	02
10	Write a program to : (iii) Find Fibonacci series for given number. (iv) Write a program to produce the following output: <pre> 1 2 3 4 5 6 7 8 9 10 </pre>	III	02
11	Array: Develop a Program to: (i) Sort list of 10 numbers. (ii) Perform addition of 3x3 matrix.	IV	02
12	Structure: Develop a Program to: (i) Create a structure called library to hold details of a book viz. accession <u>number</u> , title of the book, author name, price of the book, and <u>flag</u> indicating whether book is issued or not. Fetch some sample data and display the same. (ii) Develop and execute C Program to Add Two Distances <u>given in kilometer-meter Using Structures</u>	IV	02
13	Library Functions: Develop Program to demonstrate: (i) Use of all String handling functions. (ii) Use of few Mathematical functions. (iii) Use of few other miscellaneous functions.	V	02
14	User Defined Functions: Develop a Program to: (i) Create a function to find GCD of given number. Call this function in a <u>program</u> .	V	02

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
	(ii) Find Factorial of given number using recursion.		
15	Pointers: Develop a Program to Print values of variables and their addresses.	VI	02
16	Develop a Program to Find sum of all elements stored in given array using pointers.	VI	02
Total			32

Note

- i. A suggestive list of PrOs is given in the above table. More such PrOs can be added to attain the COs and competency. All the above listed practical need to be performed compulsorily, so that the student reaches the 'Applying Level' of Blooms's 'Cognitive Domain Taxonomy' as generally required by the industry.
- ii. The 'Process' and 'Product' related skills associated with each PrO are to be assessed according to a suggested sample given below:

S. No.	Performance Indicators	Weightage in %
a.	Correctness of algorithm	40
b.	Debugging ability	20
c.	Quality of input and output displayed (messaging and formatting)	10
d.	Answer to sample questions	20
e.	Submit report in time	10
Total		100

The above PrOs also comprise of the following social skills/attitudes which are Affective Domain Outcomes (ADOs) that are best developed through the laboratory/field based experiences:

- Follow safety practices.
- Practice good housekeeping.
- Demonstrate working as a leader/a team member.
- Maintain tools and equipment.
- Follow ethical practices.

The ADOs are not specific to any one PrO, but are embedded in many PrOs. Hence, the acquisition of the ADOs takes place gradually in the student when s/he undertakes a series of practical experiences over a period of time. Moreover, the level of achievement of the ADOs according to Krathwohl's 'Affective Domain Taxonomy' should gradually increase as planned below:

- 'Valuing Level' in 1st year
- 'Organising Level' in 2nd year
- 'Characterising Level' in 3rd year.

7. MAJOR EQUIPMENT/ INSTRUMENTS REQUIRED

The major equipment with broad specification mentioned here will usher in uniformity in conduct of experiments, as well as aid to procure equipment by authorities concerned.



S. No.	Equipment Name with Broad Specifications	Exp. S. No.
1	Computer system (Any computer system with basic configuration)	All
2	'C' Compiler	

8. UNDERPINNING THEORY COMPONENTS

The following topics/subtopics is to be taught and assessed in order to develop UOs for achieving the COs to attain the identified competency:

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
Unit – I Program Logic Developm ent	1a. Write Pseudo program logic for the given problem. 1b. Identify the given symbols of a flow chart. 1c. Explain guidelines for preparing flowchart with example. 1d. Create flowchart to logically solve the given problem.	1.1 Fundamentals of algorithms: Notion of an algorithm. Pseudo-code conventions like assignment statements and basic control structures. 1.2 Algorithmic problems: Develop fundamental algorithms to solve simple problems such as: (i) solve simple arithmetic expression (ii) find the greatest of three numbers (iii) determine whether a given number is even or odd (iv) determine whether a given number is prime. 1.3 Flowchart: Flowchart. Symbols of flowchart. Guidelines for preparing Flowchart
Unit– II Basics of C program ming	2a. Identify the given building block of a C program. 2b. Write simple 'C' program using the given arithmetic expressions 2c. Write a simple 'C' Program demonstrating the given data type conversion 2d. Write I/O Statements for the given data.	2.1 Introduction to C: History of 'C' General Structure of a 'C' program: Header files, 'main' function. 2.2 Data Concepts: Character set, tokens, keywords, Identifiers, Variables, Constant, data types, C operators, Arithmetic operators, Arithmetic expression, declaring variables, and data type conversion. 2.3 Basic Input output: Input and Output statements, using printf() and scanf(), character input/output statements, Input/output formatting, Use of comments
Unit– III Control Structure s	3a. Write a 'C' program using decision making structure for two-way branching to solve the given problem. 3b. Write a 'C' program using	3.1 Decision making and branching: Relational and logical operators, if statement, if else statement, nested if-else, if-else ladder The switch statement

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
	decision making structure for multi-way branching to solve the given problem. 3c. Write a 'C' program using loop statements to solve the given iterative problem. 3d. Use related statements to alter the program flow in the given loop.	3.2 Looping: While loop, Do... While loop For loop, Go to statement, Use of break and continue statements
Unit-IV Array and Structure	4a. Write statements to read, write the given array. 4b. Manipulate the given array of characters and numbers. 4c. Use the structure for solving the given problem. 4d. Write a sample program to demonstrate use of the given enumerated data type.	4.1 Characteristics of an array, One dimension and two dimension arrays 4.2 Array declaration and Initialization 4.3 Array of characters, Operation on array 4.4 Character and String input/output 4.5 Introduction and Features of Structures. Declaration and Initialization of Structures 4.6 Type def. Enumerated Data Type, using structures in C Program
Unit–V Function s	5a. Use the given Library function. 5b. Develop relevant user defined functions for the given problem. 5c. Write 'C' codes to pass the given function parameters using "call by value" and "call by reference" approach. 5d. Write recursive function for the given problem.	5.1 Concept and need of functions 5.2 Library functions: Math functions, String handling functions, other miscellaneous functions. 5.3 Writing User defined functions, scope of variables. 5.4 Parameter passing: call by value, call by reference. 5.5 Recursive functions
Unit-VI Pointers	6a. Use pointers to access memory locations using pointer to solve the given problem. 6b. Use pointers for performing the given arithmetic operation. 6c. Develop a program to access elements of the given array using pointers. 6d. Develop a program to access elements of the given structure using pointers.	6.1 Concepts of pointers: declaring, initializing, accessing. 6.2 Pointer arithmetic. 6.3 Handling arrays using pointers 6.4 Handling functions using pointers 6.5 Handling structures using pointers

Note: To attain the COs and competency, above listed UOs need to be undertaken to achieve the 'Application Level' of Bloom's 'Cognitive Domain Taxonomy'



9. SUGGESTED SPECIFICATION TABLE FOR QUESTION PAPER DESIGN

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	Program Logic Development	04	02	02	04	08
II	Basics of C programming	06	02	04	04	10
III	Control Structures	10	02	02	08	12
IV	Arrays and Structures	12	02	02	10	14
V	Functions	10	02	04	08	14
VI	Pointers	06	02	02	08	12
Total		48	12	16	42	70

Legends: R=Remember, U=Understand, A=Apply and above (Bloom's Revised taxonomy)

Note: This specification table provides general guidelines to assist student for their learning and to teachers to teach and assess students with respect to attainment of UOs. The actual distribution of marks at different taxonomy levels (of R, U and A) in the question paper may vary from above table.

10. SUGGESTED STUDENT ACTIVITIES

Other than the classroom and laboratory learning, following are the suggested student-related *co-curricular* activities which can be undertaken to accelerate the attainment of the various outcomes in this course:

- Prepare journal of practicals.
- Undertake micro-projects.

11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES (if any)

These are sample strategies, which the teacher can use to accelerate the attainment of the various learning outcomes in this course:

- Massive open online courses (*MOOCs*) may be used to teach various topics/sub topics.
- 'L' in item No. 4 does not mean only the traditional lecture method, but different types of teaching methods and media that are to be employed to develop the outcomes.
- About 15-20% of the topics/sub-topics which is relatively simpler or descriptive in nature is to be given to the students for *self-directed learning* and assess the development of the COs through classroom presentations (see implementation guideline for details).
- With respect to item No.10, teachers need to ensure to create opportunities and provisions for *co-curricular activities*.
- Guide student(s) in undertaking micro-projects.
- Demonstrate students thoroughly before they start doing the practice.
- Encourage students to refer different websites to have deeper understanding of the subject.
- Observe continuously and monitor the performance of students in Lab.

12. SUGGESTED MICRO-PROJECTS

Only one micro-project is planned to be undertaken by a student assigned to him/her in the beginning of the semester. S/he ought to submit it by the end of the semester to develop the industry oriented COs. Each micro-project should encompass two or more COs which are in fact, an integration of PrOs, UOs and ADOs. The micro-project could be industry application based, internet-based, workshop-based, laboratory-based or field-based. Each student will

have to maintain dated work diary consisting of individual contribution in the project work and give a seminar presentation of it before submission. The total duration of the micro-project should not be less than 16 (*sixteen*) student engagement hours during the course.

In the first four semesters, the micro-project could be group-based. However, in higher semesters, it should be individually undertaken to build up the skill and confidence in every student to become problem solver so that s/he contributes to the projects of the industry. A suggestive list is given here. Similar micro-projects could be added by the concerned faculty:

- Prepare sample mark sheet for 10 students.
- Generate salary slips of employees in an organization.
- Develop book issue system of library.
- Any other micro-projects suggested by subject faculty on similar line.

(Use structure and other features of 'C' to develop above listed applications)

13. SUGGESTED LEARNING RESOURCES

S. No.	Title of Book	Author	Publication
1	Programming in ANSI C	Balgurusamy, E.	McGraw Hill Education, New Delhi 2012, ISBN: 978-1259004612
2	The C Programming Language	Brian, W. Kernighan, Ritchie Dennis	PHI Learning Private Limited, New Delhi 1990, ISBN: 978-8120305960
3	Let us C	Kanetkar, Yashawant	BPB Publications, New Delhi 2016, ISBN:978-8183331630
4	C Programming	Dr. Rajendra Kawale	Devraj Publications, Dist Solapur, Maharashtra

14. SOFTWARE/LEARNING WEBSITES

- <http://nptel.ac.in/courses/106105085/4>
- www.w3schools.com
- www.programiz.com/c-programming
- <https://www.codecademy.com/courses/getting-started-v2/0/1>
- <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-087-practical-programming-in-c-january-iap-2010/>
- <http://spoken-tutorial.org/>

