

Program Name : Electronics and Telecommunication and Digital Electronics
Program Code : EJ/DE/ET/EN/EX
Semester : Second
Course Title : 'C' Programming Language
Course Code : 22218

1. RATIONALE

Automation Industry needs to build Microcontroller based applications which are being developed using 'C'. This course deals with concepts of programming to enhance programming skills of diploma students. This course will enable the students to inculcate programming concepts and methodology to solve engineering problems.

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 electronic engineering 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:

- Interpret the basic code of 'C'.
- Implement decision making in 'C' programming.
- Use Arrays and string in 'C' programming.
- Use functions in 'C' programs for modular programming approach.
- Use pointers to increase efficiency of programs.
- Implement basic concept of structure in 'C'.

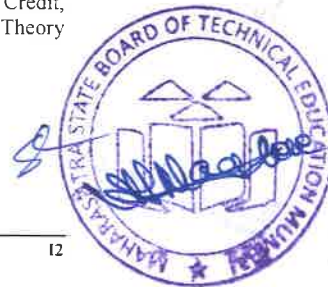
4. TEACHING AND EXAMINATION SCHEME

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
4	-	4	8	3	70	28	30*	00	100	40	50@	20	50	20	100	40

(*): 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, '#': No Theory Examination

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



5. 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.

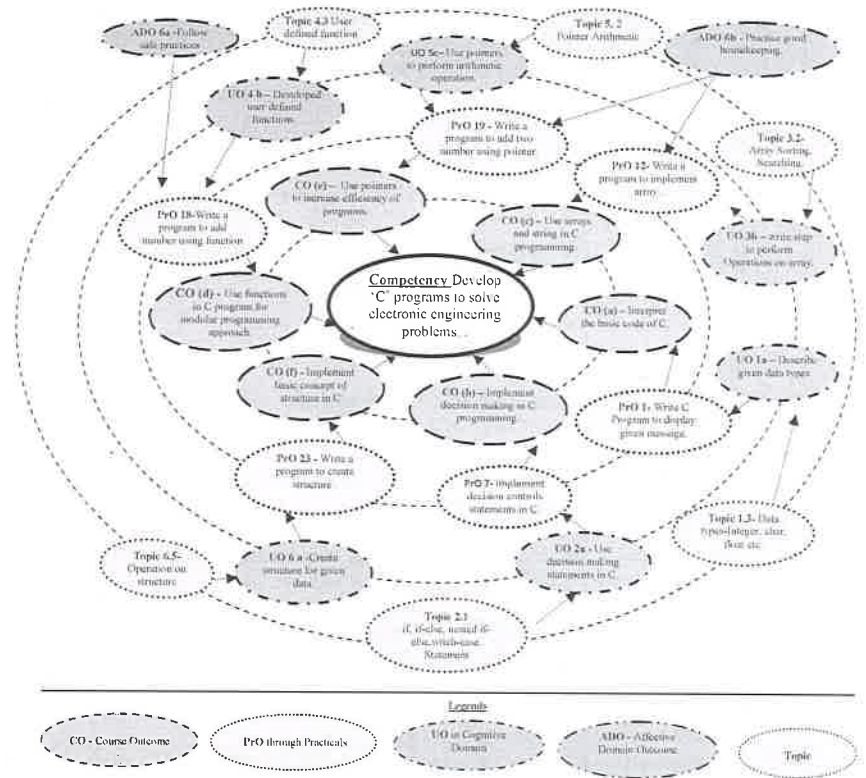


Figure 1 - Course Map

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	a) Write a 'C' program to display hexadecimal, decimal, octal format of entered number using %d, %c, %i, %f, %g, %u, %o, %s, %x b) Write algorithm and draw flow chart for following problems: i. Addition of two numbers ii. Exchange value of two variable	I	02

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
2	Write a program to perform following operations: (a) Display the message "Hello World", name, address, date of birth and email_id using printf () function. (b) Logical operations: & (AND), (OR) for given values, Bitwise operations :<< (LEFT SHIFT), >> (RIGHT OPERATOR) for given values.	I	02
3	(a) Write a program to display current time and date using time.h header file. (b) Write a program to display addition of value of resistor R, Where, i. R series =R1 + R2+R3 and ii. R parallel =1/ R1 + 1/ R2+1/R3 <i>Note. Use math.h header file.</i>	I	02*
4	(a) Write a program to calculate inductive resistance (F_L) with the help of given formula $F_L = 2 * \pi * f * L$. Where π , f, L are given data. (b) Write a program to calculate capacitive resistance (F_C) with the help of given formula $F_C = 1/(2 * \pi * f * C)$. Where π , f, C are given data. <i>Note. Develop above programs using local variables, global variables and arithmetic operators.</i>	I	02
5	Implement decision control statements in C using 'if' (a) Write a program to find whether given number is even or odd. (b) Write a program to find whether given number is Positive, negative or zero.	II	02*
6	(a) Write a program to find the largest among n numbers using 'if-else'. (b) Write a program to determine leap year using 'if-else'.	II	02
7	Implement decision control statements in 'C' using 'nested if-else' (a) Determine whether a string is palindrome. (b) Find the greatest of the three numbers using conditional operators.	II	02
8	Write a program to perform addition, subtraction; multiplication and division according to user's choice using switch case statement for given data	II	02
9	Implement loop control statements in 'C' using 'for' loop (a) Write a program to print the table for given no. in one column. (b) Write a program to count the number of digit in a given number.	II	02
10	Implement loop control statements in 'C' (a) Find Fibonacci series for given number. (b) Write a program to produce the following output: <pre> 1 2 3 4 5 6 7 8 9 10 </pre>	II	02
11	(a) Print the Result sheet: Conditions given are: marks >=40%	II	02



S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
	pass, marks <40 % fail. marks 60>=first class, marks above 75 % distinction, marks >100 and marks < 0 not valid		
12	(a) Write a program to declare, modify and print elements of a given data array. (b) Write a program to find highest marks in a class of n students using array.	III	02*
13	(a) Write a program to copy of one array into second array for given data elements (b) Write a program to create an array by reversing the elements of the given array.	III	02
14	(a) Write a program to sort numbers in ascending and descending in a given array. (b) Write a program to add two matrices of size 3*3 store additions in third matrix for given data elements.	III	02
15	(a) Write a program that accept a string from user and print that string (b) Write a program that accept a string and compare it with existing string.	III	02*
16	(a) Write a program to accept and concatenate two strings. (b) Write a program to find length of a string.	III	02
17	Library Functions: Develop Program to demonstrate: (a). Use of all String handling functions. (b). Use of few Mathematical functions.	IV	02*
18	(a) Write a program to add two numbers using function. (b) Write a program to perform addition, subtraction, multiplication and division using switch case statement and user defined function for given data	IV	02*
19	(a) Write a program to use address operator (&) and pointer operator (*) for given data (b) Write a program to add two integer numbers using pointer.	V	02*
20	(a) Write a program to calculate the sum of elements of given array using pointer. (b) Write a program to access the array elements using pointer.	V	02
21	(a) Write a program to interchange given values of two variables using call by value mechanism. (b) Write a program to interchange given values of two variables using call by reference mechanism.	V	02*
22	Write a program to exchange given values of two variables using pointer.	V	02
23	Create structure DATE using 'C' having members' day, month, year and assign initial values to that structure.	VI	02
24	Write a program to create a structure for student having data members like Roll No., Name, Class, marks in three subjects and calculate the % of marks.	VI	02
Total			48

- i. A suggestive list of **PrOs** is given in the above table. More such PrOs can be added to attain the COs and competency. A judicious mix of minimum 12 or more practical need to be performed. out of which, the practicals marked as '*' are compulsory, so that the student reaches the 'Precision Level' of Dave's 'Psychomotor Domain Taxonomy' as generally required by the industry.
- ii. The 'Process' and 'Product' related skills associated with each PrO is to be assessed according to a suggested sample given below:

S.No.	Performance Indicators	Weightage in %
1	Write algorithm and draw flow chart.	20
2	Use 'C' software tool for programming to create, edit, compile the 'C' programs/applications	40
3	Debug, test and execute the programs/applications	20
4	Able to answer oral questions.	10
5	Submission of 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:

- Handle command prompt environment.
- Experiment with C / C++ environment.
- Plan, construct, compile, debug and test C programs.
- 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.

1. MAJOR EQUIPMENT/ INSTRUMENTS REQUIRED

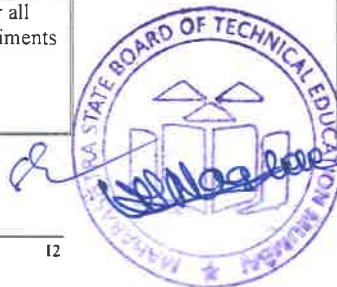
The major equipment with broad specification mentioned here will usher in uniformity in conduct of experiments, as well as aid to procure equipment by authorities concerned.

S. No.	Equipment Name with Broad Specifications	Expt. S. No.
1	Hardware: Personal computer, (i3-i5 preferable), RAM minimum 2 GB onwards.	For all Experiments
2	Operating system: Windows XP/Windows 7/LINUX onwards.	
3	Software: Turbo C , or Microsoft Visual Studio 2005 onwards (Optional).	

2. 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 Overview of C Programming	1a. Describe the given data type. 1b. Construct algorithm, flow chart for the given problem. 1c. Use pre-increment and post-increment operators in the given situation. 1d. Use bitwise operators in the given situation.	1.1 Structure of 'C' program. Assembler, Linker, Compiler, Interpreter. 1.2 'C' character set-keywords, identifiers, types of constants (Integer, single character, string, and real) variables, scope of variables, concept of ASCII. 1.3 Data types: integer- unsigned, signed, long, float- float, double, character-char, string, octal, hexadecimal 1.4 Algorithm and flow chart. 1.5 Formatted input and output statements. Input and output function. 1.6 Operators and expressions: a. Operators in 'C'- arithmetic, logical, assignment, relational, increment and decrement, conditional, bit wise, special operators b. Expressions c. Precedence and associativity.
Unit– II Decision control and Loop control	2a. Write a 'C' program using the given decision making structure for two-way branching. 2b. Write a 'C' program using the decision making structure for multi-way branching. 2c. Write a 'C' program using loop statements to solve the given iterative problem. 2d. Use related statements to alter the program flow in the given loop.	2.1 Decision making if statement (if, if-else, nested if-else), switch –case statement. 2.2 Repetition in 'C' (loop control statement) while, do-while and for loop, break and continue statement, nested loops.
Unit– III Array and Strings	3a. Write steps to access elements of the given array. 3b. Write steps to perform operation on the given array. 3c. Write steps to initialization and declaration of the given string in 'C' program. 3d. Apply relevant control statement on the given strings to manipulate its elements.	3.1 Introduction to Array and its types 3.2 Declaration, initialization of array, accessing elements of an array, adding, deleting, sorting & searching. 3.3 Introduction to string Initializing, declaring and display of string 3.4 String handling functions from standard library (strlen (), strcpy (), strcat (), strcmp(), strlen(),strupr()):



Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
Unit-IV Functions	4a. Use inbuilt functions for the given problem. 4b. Develop relevant user defined functions for the given problem. 4c. Write 'C' codes to Pass function parameters using the given approach. 4d. Write recursive function for the given problem.	4.1 Concept and need of functions 4.2 Library functions: Math functions, String handling functions, other miscellaneous functions. 4.3 Writing User defined functions, scope of variables, 4.4 Parameter passing: call by value, call by reference. 4.5 Recursive functions
Unit-V Pointers	5a. Use pointer for address access to manipulate the given data. 5b. Use pointers to access memory locations to solve the given problem. 5c. Use pointers for performing the given arithmetic operation. 5d. Develop a program to access array elements using the given pointers.	5.1 Concept of pointer and pointer variables, initialization of pointer, call-by-reference. 5.2 Pointer arithmetic. 5.3 Handling arrays using pointers 5.4 Handling functions using pointers
Unit-VI Structures	6a. Create a structure for the given data. 6b. Develop a program to access elements of structure using pointers. 6c. Use the structure for solving the given problem. 6d. Use of enumerated data type in structure to solve the given program.	6.1 Introduction and Features and Syntax of structure 6.2 Declaration and Initialization of Structures 6.3 Initializing, accessing structure members using pointers 6.4 Type def, Enumerated Data Type, using structures in C Program 6.5 Operations on structure.

Note: To attain the COs and competency, above listed UOs need to be undertaken to achieve the 'Application Level' and above 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	Overview of C Programming	08	02	04	04	10
II	Decision control and Loop control	12	04	04	04	12
III	Array and Strings	14	04	06	06	16
IV	Functions	12	02	04	06	12
V	Pointers	10	02	04	04	10
VI	Structures	08	02	04	04	10
Total		64	16	26	28	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.

This specification table also provides a general guideline for teachers to frame internal end semester practical theory exam paper which students have to undertake.

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 journals based on practical performed in laboratory.
- Give seminar on relevant topic.
- Library/E-Book survey regarding 'C' used in electronics industries.
- Prepare power point presentation or animation for showing different types of 'C' applications.
- Find and Utilize android applications related to 'C'.
- Undertake a market survey of different 'C' application and compare with the following points.
 - Available applications.
 - Application profile.

11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES (if any)

These are sample strategies, which the teacher can use to accelerate the attainment of the various 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.
- No. of practical's selection to be performed should cover all units.

12. SUGGESTED MICRO-PROJECTS

Only one micro-project is planned to be undertaken by a student assigned to him/her in the 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:

- a. **Modern Periodic Table using 'C'** - Each group will prepare a periodic table using functions 'Void add()' and 'Void show()'
- b. **Simple Calculator** - Each batch will prepare a menu driven program to perform any five mathematical operations.
- c. **Employee Record System** - Each batch will prepare a menu driven program to perform following operations :
 - i. Add record
 - ii. List record
- d. **Digital clock using 'C'**
- e. **String Manipulation project** - Each batch will prepare a menu driven program to perform following operations (any five) :
 - i. Substrings
 - ii. Palindromes
 - iii. Comparison
 - iv. Reverse string
 - v. String to integer
 - vi. Sort a string
- f. **Matrix Operations** - Each batch will prepare a menu driven program to perform following operations:
 - i. Matrix addition
 - ii. Matrix multiplication
 - iii. Matrix transpose
 - iv. Sum of diagonal of a matrix.
- g. **Basic mathematic functions** - Each batch will prepare a menu driven program to perform following operations:
 - i. Pascal triangle
 - ii. Armstrong No.
 - iii. Floyd's triangle
 - iv. HCF and LCM.
- h. **Patterns** - Each batch will prepare a menu driven program to obtain following patterns (any three):

```

1          1          *          1
121       12        **         22
12321    123       ***        333
1234321  1234     **         4444
          *

```

13. SUGGESTED LEARNING RESOURCES

S. No.	Title of Book	Author	Publication
1	Programming in 'C'	Balguruswamy, E.	Tata McGraw Hill May 2012, New Delhi ISBN:978-1-25-900461-2.
2	Let us 'C'	Kanetkar, Yashwant	BPB Publication July 2016, New Delhi. ISBN : 9788183331630,
3	Basic computation and programming with 'C'	Saha, Subrata ; Mukherjee, Subhodip	Cambridge 2016, New Delhi. ISBN: 978-1-316-60185-3

14. SOFTWARE/LEARNING WEBSITES

- a. Turbo C Editor
- b. Dosbox
- c. www.tutorialspoint.com/cprogramming
- d. www.cprogramming.com
- e. www.sourcecodesworld.com/source/LanguageHome.asp?LangId=1
- f. <http://fresh2refresh.com/c-programming/c-basic-program/>
- g. <http://www.c4learn.com/c-programs/>
- h. <http://computer.howstuffworks.com/c2.htm>
- i. <http://www.programiz.com/c-programming/examples>
- j. www.indiastudycenter.com/studyguides/cs/default.asp
- k. Android application resources for 'C' programming from Google Play store.

