

Government of Karnataka
Department of Technical Education
Board of Technical Examinations, Bengaluru

Course Title	: C- Programming Lab	Course Code	: 15EC37P
Semester	: Third	Credits	: 3 Credits
Teaching Scheme in Hrs (L:T:P)	: 0:2:4	Course Group	: Core
Type of course	: Tutorial + Practical	Total Contact Hours	: 78
CIE	: 25 Marks	SEE	: 50 Marks

Prerequisite

Basics of digital electronics and number systems, and logical thinking

Course Objectives

1. Understand the need for high-level languages, particularly C, and programming paradigms.
2. Familiarize with program-development environment and development of simple programs in C
3. Understand the need and role of data-structures such as arrays, strings and pointers, and functions in C-program development.

Course Outcomes

At the end of the course, the students will be able to attain the following COs

Course Outcome		CL	Experiments linked	Linked PO	Teaching Hrs
CO1	Appreciate the necessity and evolution of the computer language and selection of programming paradigms.	R/U/A	Unit-1 Chapter 1	1,2,4,10	5
CO2	Perform algorithm development, its representation using flow-charts and analysis of solution for simple problems.	R/U/A	Unit 1 Chapter1, 2,3,4,5,6 and practice programs	1,2,3,4,10	31
CO3	Employ C-programming language for simple real-world problem-solving using computer.	R/U/A	Unit 2 Programs 1 to 6	1,2,3,4,5, 10	18
CO4	Develop and analyze simple programs/functions and their integration in larger problem solving.	R/U/A	Unit 2 Programs 7 to 11	1,2,3,4,5, 10	18
Two CIE/IA Tests					06
Total					78

Legend: E- Experiment, R-Remember, U-Understand, A-Application, CL-Cognitive Level, PO-Program Outcome

Note: Total sessions include two tests

Course-PO Attainment Matrix

Course	Programme Outcomes									
	1	2	3	4	5	6	7	8	9	10
C- Programming Lab	3	3	3	3	3	--	--	--	--	3
<p>Level 3- Highly Addressed, Level 2-Moderately Addressed, Level 1-Low Addressed.</p> <p>Method is to relate the level of PO with the number of hours devoted to the COs which address the given PO.</p> <p>If $\geq 40\%$ of classroom sessions addressing a particular PO, it is considered that PO is addressed at Level 3</p> <p>If 25 to 40% of classroom sessions addressing a particular PO, it is considered that PO is addressed at Level 2</p> <p>If 5 to 25% of classroom sessions addressing a particular PO, it is considered that PO is addressed at Level 1</p> <p>If $< 5\%$ of classroom sessions addressing a particular PO, it is considered that PO is considered not-addressed.</p>										

Course Contents

Unit-1: Tutorials and Practice Exercises

36 Hours

Tutorial		
Sl. No.	Topic/Exercises	Duration (Hr.)
1	<p>Introduction Definition, need, and types of programming languages and their selection criterion.</p> <ul style="list-style-type: none"> • Need for programming language • Computer programming languages: Machine language, assembly/low-level language, middle-level language and high-level languages. Features and advantages and disadvantages. • Algorithm and methods of writing algorithms: pseudo-code and flow-charts • Features of pseudo-code or flow-charts • Programming paradigms: Features of procedure oriented programming (POP) and object oriented programming (OOP) paradigms • Program development process (writing, editing, compilation, linking, execution and debugging and file extensions) and tools • Programming errors: Syntax, semantic, linker, and run-time (logical and data) errors • C-program format • Introduction to standard input and output statements • Variables and constants: Definition, naming (identifiers or labels for different entities), initialization and accessing of variables. Constants and their representation • Data types: classification, memory requirement, range of values, usage and type specifiers 	10

	<ul style="list-style-type: none"> • Operators and Operands: Unary, binary and ternary operators. Arithmetic, logical, relational, combinational-assignment and special operators. Precedence and associativity. Unary and binary operands • Statements- tokens and expressions • Type casting-automatic and forced • Escape characters • I/O statements in detail 	
2	Control structure <ul style="list-style-type: none"> • Branching: Conditional (<i>if, if-else</i>, nested and ladder <i>if-else, switch</i> constructs) and unconditional (<i>break, continue</i> and <i>goto</i> statements) • Looping: Entry-controlled (<i>for</i> and <i>while</i>) and exit-controlled (<i>do while</i>) loops 	06
3	Arrays/Matrices, strings and pointers <ul style="list-style-type: none"> • Definition, declaration, initialization (static and run-time or dynamic) of arrays, strings and pointers • Accessing of strings, arrays (one and two dimensional), and pointers 	06
4	Functions <ul style="list-style-type: none"> • Concept, pros and cons, classification, creation and application of functions • Parameter passing: Pass by value and address methods • Library and user-defined functions • Recursive and non-recursive functions • Data visibility and longevity 	05
5	User-defined data-type <ul style="list-style-type: none"> • Definition, declaration, initialization of members and variables of structures and unions • Distinction between structures and unions • Accessing of members of structures and unions 	06
6	Macros Definition, classification and application of macros in program development	03
Total Duration (Hr.)		36
Practice Exercises		
1.	Write algorithm, flow-chart and program for the following problems (Assume relevant data if necessary and right units. Search and use library functions wherever possible). <ol style="list-style-type: none"> 1. Display your country name and beep at the end. 2. Display the following text: C was developed by “Dennis M Ritchie” in 1972 3. Compute the area and circumference of a circle given the radius 4. Compute simple interest given the interest rate, principal and duration 	

	<ol style="list-style-type: none"> 5. Compute compound interest given the interest rate, principal, compounding-nature and duration 6. Swap contents of two variables without using intermediate variables 7. Display expenditure incurred against each item, such as fee receipt 8. Factorial of a single digit number 9. Absolute value of a number 10. Verify if the given number is prime or not 11. Largest of three numbers using if-else and ternary operators 12. Logarithm of a number. 13. $\sin(\theta_1 + \theta_2) + \cos(\theta_1 - \theta_2)$ given θ_1 and θ_2 in degrees 14. Average of N numbers read through keyboard (i.e., at run-time) 15. Average of the numbers in an $N \times M$ matrix 16. Search a number in an array of N integers 17. Verify Pythagoras theorem. 18. Compute the area and circumference of a circle given the radius using macro for π. 19. Find smallest of three numbers using macro 20. Use function to swap contents of two variables using (a) by <i>pass-by-value</i> and (b) <i>pass-by-address</i> methods
2	<p>Analysis of the program</p> <p>Given the program or block of program code, analyze the same and record the output or error, as the case may be. Justify your answer. Examples,</p> <ol style="list-style-type: none"> 1. <pre>#include<stdio.h> main() { int k=5, l=24; printf("%f", k/l); }</pre> 2. <pre>#include<stdio.h> main() { printf("Saying by Mahatma\tGhandiji: \nWhat can\'t be tolerated must be endured"); }</pre> 3. <pre>int i; for (i=1; i<10; i--);</pre>

Unit – 2: Graded Exercises

42 Hours

Write the algorithm/flow-chart and code for the following problems

Note: Assume appropriate data and units wherever necessary. Library functions, unless specified, can be used wherever necessary.

Sl. No.	Graded Exercises	Duration (Hr.)
1	Given the resistance and standard tolerance, generate the color bands of the resistor	3
2	Given the total marks, generate the result-class of a student	3

3	Given an alphabet, display whether it is VOWEL or CONSONANT	3
4	Multiply two matrices using array-subscript method	3
5	Add two matrices using pointers	3
6	Given a string, reverse it using your own code and a library function; compare the time consumed in both approaches	3
7	Given x in radians, compute $\sin(x)$ using series method and the corresponding library function; compare both methods for execution time.	3
8	Create a structure to represent a <i>book</i> and declare an array of N variables of the type <i>book</i> and sum and display the cost members of all the book variables	3
9	Simulate the arithmetic calculator	3
10	Sort a list of N numbers using user defined function	3
11	Two open-ended experiments of similar nature and magnitude of the above are to be assigned by the teacher (Student is expected to solve and execute/simulate independently. Open-ended programs covered here shall not be included in the exam)	6
	Two Internal Assessment Tests	6
	Total	42

Unit – 3: Student Activity for CIE

Sl. No.	Activity
1	Prepare/Collect notes pertaining to Unit-1 and solve practice programs specified in Unit-1
2	Collect or write any 10 programs (open-ended) not specified in Unit-1 or Unit-2
3	Prepare a list of high-level programming languages, collect features of any five such high-level programming languages and collect/prepare material on C-program development tools and their usage.

Institutional Activity (No marks)

The following are suggested institutional activities, to be carried out at least one during the semester. The course teacher/coordinator is expected to maintain the relevant record (Containing, Activity name, Resource persons and their details, duration, venue, student feedback, etc) pertaining to Institutional activities

Sl. No.	Activity
1	Organize hands-on practice on installation of Compilers and use of IDE for program development.
2	Organize seminar on free-open source software
3	Conduct quiz on C programming

References

1. C Programming By Balaguruswamy, TMH Publishers, ISBN-10: 8131716813, 2009.
2. C Programming By Kernighan and Dennis Ritchie

Course Delivery

The idea behind this course delivery is to provide relevant tutorial and hands-on practice concurrently. The course will be normally delivered through two-hour tutorials and four-hour hands-on practice per week; hands-on practice shall include practice exercises and graded exercises. Normally, one-hour tutorial followed by two-hour hands-on practice is recommended in each class. In Unit-1, tutorials and practice may be carried out concurrently. However, graded exercise (Unit-II) can also be covered at appropriate point of tutorials of Unit-1. Activities are carried-out off class.

Course Assessment and Evaluation Scheme

Master Scheme

Assessment Method	What		To Whom	Assessment mode /Frequency /timing	Max. Marks	Evidence Collected	Course Outcomes
Direct assessment	CIE	IA	Students	Two tests ⁺	10	Blue Books	1 to 4
				Record [@]	10	Record Book	1 to 4
				Activity [*]	05	Report/Sheets	1 to 4
	SEE	End exam		End of the course	50	Answer Scripts at BTE	1 to 4
				Total	75		
Indirect assessment	Student feedback on course		Students	Middle of the Course	Nil	Feedback Forms	1 to 4 & Delivery of course
	End of course survey			End of the Course	Nil	Questionnaires	1 to 4, Effectiveness of delivery instructions & assessment methods

Legends: CIE-Continuous Internal Evaluation, SEE- Semester End-exam Evaluation

1. I.A. test shall be conducted as per SEE scheme of valuation. However obtained marks shall be reduced to 10 marks. Average marks of two tests shall be rounded off to the next higher digit.
2. Rubrics to be devised appropriately by the concerned faculty to assess Student activities.

*Students should do activity as per the list of suggested activities/ similar activities with prior approval of the teacher. Activity process must be initiated well in advance so that it can be completed well before the end of the term.

@ Record Writing: Average of marks allotted for each experiment; fractional part of average shall be rounded-off to next higher integer.

Composition of CLs

Sl. No.	Cognitive Levels (CL)	Weightage (%)
1	Remembering	20
2	Understanding	30
3	Applying	50
Total		100

Continuous Internal Evaluation (CIE) pattern

(i) **Student Activity (5 marks):** The student activities in Unit-3 or similar activities can be assigned by the teacher

Execution Notes:

- Each batch of 2 students is assigned at least one activity listed in Unit-3 based on interest of the students. Student can also choose any other similar /relevant activity with prior approval from the concerned teacher.
- Teacher is expected to observe and record the progress of students' activities
- Assessment is made based on quality of work as prescribed by the following **rubrics** table

(ii) Model of rubrics for assessing student activity (for every student)

Dimension	Scale					Marks (Example)
	1 Unsatisfactory	2 Developing	3 Satisfactory	4 Good	5 Exemplary	
1. Research and gathering information	Does not collect information relate to topic	Collects very limited information, some relate to topic	Collects basic information, most refer to the topic	Collects more information, most refer to the topic	Collects a great deals of information, all refer to the topic	3
2. Full-fills team roles and duties	Does not perform any duties assigned to the team role	Performs very little duties	Performs nearly all duties	Performs almost all duties	Performs all duties of assigned team roles	2
3. Shares work equality	Always relies on others to do the work	Rarely does the assigned work, often needs reminding	Usually does the assigned work, rarely needs reminding	Always does the assigned work, rarely needs reminding.	Always does the assigned work, without needing reminding	5
4. Listen to other team mates	Is always talking, never allows anyone to else to speak	Usually does most of the talking, rarely allows others to speak	Listens, but sometimes talk too much,	Listens and talks a little more than needed.	Listens and talks a fare amount	3
Total marks						ceil(13/4)= 4

(iii) CIE/IA Tests (10 Marks)

Two tests shall be conducted in accordance with SEE pattern and the marks shall be scaled down to 10. Average of two tests, rounding-off any fractional part thereof to next higher integer, shall be considered for CIE/IA.

(iv) Record Evaluation (10 Marks)

Every experiment shall be given marks, in the scale of 10, after its conduction based on student's performance and quality of write-up. Average of them, by rounding-off any fractional part thereof to next higher integer, shall be considered for CIE/IA.

Semester End-exam Evaluation (SEE) Scheme

Sl. No.	Scheme	Max. Marks
1	Short questions on Unit-1 (only write-up)	05
2	Writing steps /Algorithm/Procedure (20% weightage) and program (80% weightage) for two questions from the graded exercises	20
3	Execution/Implementation of either one of the programs written given in 2	15
4	Open-ended problem: Writing program and its execution/implementation.	05
5	Viva-voce	05
TOTAL		50
Note:		
1. Candidate is expected to submit record for the examination.		
2. Student shall be allowed to execute the program even if she/he is unable to write the procedure/steps/algorithm.		
3. Open-ended problem is of the nature and magnitude similar to graded exercises in Unit-2, and it can be assigned by the examiner. Further, open-end programs executed in 11 of Unit-2 shall be excluded. Idea behind open-end program is to assess the ability of a student to write any program.		

Model Questions for Practice and Semester End Examination

Graded Exercises

1. Write a C program that computes resistance of any resistor given its colour codes Red-Yellow-Blue and tolerance code Gold.
2. Write a C program to generate result-class of a set of students given their individual total marks

Sl No	Reg. No.	Name of the student	Marks obtained out of 600	Result
1		ABC	456	?
2		ABD	232	?
3		XYZ	345	?
4		MNO	577	?
5		PXY	365	?

3. Write a C program to count number of vowels in any given word
4. Write a C program to multiply two matrices
5. Write a C program to add two 3x4 matrices
6. Write a C program to reverse a given word

7. Write a C program to compute $\sin(x)$ given x in radians using series method; relevant library functions can be used
8. Write a C program containing a structure to represent book; create five such books and compute the sum of the costs of all the books
9. Write a C program to simulate an arithmetic calculator that can perform addition, subtraction, multiplication and division
10. Write a C program to sort an array of N numbers using a user-defined function

End