

COURSE TITLE- 'C' PROGRAMMING
 COURSE CODE 6S204

PROGRAMME & SEMESTER

Diploma Programme in which this course is offered	Semester in which offered
COMPUTER ENGINEERING INFORMATION TECHNOLOGY	SECOND SEMESTER

1. RATIONALE

This Course intends to develop programming skills in the students, using a popular structured programming language 'C'. The students will learn step by step procedure (i.e. flowcharting & Algorithm) of any program development process. The programming skills thus acquired can be used for developing programs with advance level programming features which in turn will be helping in developing practical applications for the scientific, research and business purposes.

2. LIST OF COMPETENCIES

At the end learning this course student will be able to:

"Develop structured, modular and memory efficient programs in 'C' using arrays, functions, pointers."

3. TEACHING AND EXAMINATION SCHEME

Teaching Scheme (Hours/ Credits)			Total Credits (L+T+P)	Examination Scheme (Marks)				
				Theory		Practical		Total
L	T	P	C	ESE	PT	ESE #(PR)	PA (TW)	
03	-	04	07	80	20	25#	25	150
Duration of the Examination (Hrs)				03	01	02	--	

Legends : L-Lecture; T-Tutorial/Teacher Guided Theory Practice ; P-Practical; C- Credits;
 ESE- End Semester Examination; PT – Progressive Test, PA- Progressive Assessment, OR
 – Oral Examination, TW - Term Work, # External, @ Internal



4. COURSE OUTCOMES

At the end of learning this course students will be able to :-

1. Develop algorithm and Draw the flowchart for 'C' Programming.
2. Develop a program using decision and loop statement.
3. Implement program using array.
4. Use functions and pointer in given problem statement.
5. Create structure for different data type in one head.

5. DETAILED COURSE CONTENTS

Unit	Major Learning Outcomes (Cognitive Domain Only)	Topics and Sub-topics
UNIT-I Basics of c programming	1a. Draw flow chart to solve given problem logically. 1b. Develop Algorithm to solve given program. 1c. Comprehend general structure of 'C' program 1d. Declare and define variables 1e. Write and execute simple program in 'C' 1f. Use arithmetic, relational and logical operators for forming expressions. 1g. Format input and output using 'C' statements.	1.1 Introduction to C and General structure of 'C' program 1.2 Features and Advantages of C language. 1.3 Character set, 'C' tokens Keywords and Identifiers 1.4 Constants and Variables Data Types 1.5 Modifiers and type conversion 1.6 Input and Output statements in 'C' 1.7 Types of Operators and Expression: Arithmetic, Relational, Assignment, Logical, conditional operators and expressions, Write, compile, execute a simple 'C' program
UNIT-II Control and loop statements	2a. Develop programs using decision making statements in 'C' language.	2.1 Decision Statements 2.2 Unconditional branching: goto statement



	2b. Develop programs using structured loop control statements in 'C' language	2.3 Conditional branching statements: If statement, If-else statement, Nested If else statement 2.4 If-else-if Ladder statement 2.5 Break, continue and goto statements, switch statements 2.6 Loop Control Statements: for loop, While loop, Do-while loop
UNIT-III Introduction to Array	3a. Declare and define array. 3b. Develop programs using array in 'C' language 3c. Develop, debug and execute programs which use reading, writing and manipulating Arrays. 3d. Describe string function	3.1 Array definition and Declaration 3.2 Concept of one dimensional and two dimensional array 3.3 Accessing and initialization an array 3.4 Characteristics of an array 3.5 Introduction of String 3.6 Declaration and Initialization of String 3.7 gets(), puts() functions in <String.h> such as strlen(), strcmp(), strcpy(), strcmp(), strcat() and all.
UNIT-IV Pointer and Function	4a. Develop, debug and execute modular programs by writing and using Functions 4b. Develop, debug and execute programs using Pointers 4c. Declare and initialize pointer	4.1 Introduction and Features of Pointers 4.2 Declaration of Pointer, Pointer initialization, pointer arithmetic operation 4.3 Array using pointer and array of pointers. 4.4 Basics of Functions, Built-



		<p>in and user defined functions</p> <p>4.5 Advantages of using Functions</p> <p>4.6 Working of a Function</p> <p>4.7 Declaring, Defining and calling user defined Functions</p> <p>4.8 Types of functions: i) no return type no argument list ii) no return type with argument list iii) return type no argument list iv) return type with argument list.</p> <p>4.9 Call by Value and call by Reference and recursive function</p>
UNIT-V Structure and Union	<p>5a. Implement program for different Data types under a single structure</p> <p>5b. Describe array of structure and pointer to structure.</p> <p>5c. Describe union with its use</p> <p>5d. Utilize memory effectively using Union</p>	<p>5.1 Introduction and Features of Structures</p> <p>5.2 Definition and Declaration of Structures</p> <p>5.3 Memory allocation of structure</p> <p>5.4 Array of Structures and Pointers to Structure</p> <p>5.5 Nested structure.</p> <p>5.6 Declaration of Union, its use and how to access it</p> <p>5.7 Create dynamic memory using <malloc.h> for structure variable.</p>

Course Curriculum Design Committee

Sr No	Name of the faculty members	Designation and Institute
1	Ms. R.S.Sindge	LIT, P.L. Govt. Polytechnic, Latur
2	Ms. V.B. Kundlikar	LIT, Govt. Polytechnic, Aurangabad

(Member Secretary PBOS)

(Chairman PBOS)



10. SUGGESTED LEARNING RESOURCES

List of Books

Sr.No	Author	Title of Books	Publication
1	Kamthane A.N.	Programming in 'C'	Pearson,2012
2	Balaguruswami,E.	Programming in ANSIC	TMH,2012
3	Kanetkar, Yashavant	Let us 'C'	BPB publications,2010

11. List of Major Equipment/ Software

1. Computer System with latest configuration
2. 'C' Compiler

12. List of Software/Learning Websites

1. 'C' Programming Language: <http://www.w3schools.in/cprogramming-language/intro/>
2. Learn C Online: <http://www.learnonline.com/>
3. 'C' Frequently Asked Questions: <http://www.c-faq.com>
4. 'C' Programming: <http://www.cprogramming.com>
5. Sams Teach Yourself C in 24 Hours: <http://aelink.free.fr/c/>

13. MAPPING OF PROGRAMME OUTCOMES (POs) AND PROGRAMME SPECIFIC OUTCOMES (PSOs) WITH COURSE OUTCOMES (COs)

CO. NO.	Course Outcome	POs										PSOs	
		1	2	3	4	5	6	7	8	9	10	1	2
1	Develop algorithm and flowchart for 'C' Programming	1	-	2	3	-	-	-	-	-	-	1	1
2	Develop a program using decision and loop statement	-	-	3	3	-	-	-	-	-	-	2	2
3	Implement program using array.	-	1	2	1	-	-	-	-	-	-	2	2
4	Comprehend and use the concept of functions and pointer.	-	1	3	3	-	-	-	-	-	-	2	2
5	Create structure for different data type in one head	-	2	1	1	-	-	-	-	-	-	2	2
6	Open a file using 'C' Program.	-	1	1	1	-	-	-	-	-	-	2	2

6. SUGGESTED SPECIFICATION TABLE WITH HOURS & MARKS (THEORY)

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R	U	A	Total
I	Basics of c programming	08	3	5	6	14
II	Control and loop statements	10	3	5	8	16
III	Introduction to Array	08	4	4	6	14
IV	Pointer and Function	12	4	6	10	20
V	Structure and Union	10	4	6	6	16
Total		48	18	26	36	80

Legends: R = Remembrance; U = Understanding; A = Application and above levels (Revised Bloom's Taxonomy)

7. SUGGESTED LIST OF EXERCISES/PRACTICAL/EXPERIMENTS

The exercises/practical/experiments should be properly designed and implemented with an attempt to develop different types of skills leading to the achievement of the competency. Following is the list of exercises/practical/experiments for guidance.

Sr. No.	Unit No.	Practical Exercises (Outcomes in Psychomotor Domain)	Approx. Hrs. required
1.	1	Draw Flow Chart and write algorithm for at least four problems.	2
2.	1	i. Write programs using Constants, Variables & arithmetic expression. ii. Write program to calculate average of numbers using arithmetic operators	2
3.	1	Execute programs to create variable with different data types, Type modifiers and Type conversion.	2
4.	1	Execute programs providing insight to formatted and unformatted input and output in c	2
5.	1	Execute programs providing understanding of Relational	2

Sr. No.	Unit No.	Practical Exercises (Outcomes in Psychomotor Domain)	Approx. Hrs. required
		operators.	
6.	1	Execute programs using logical and bitwise operators.	2
7.	2	Execute programs using If, If-else, If-else-if and Nested If statements.	4
8.	2	Execute programs using break, continue, goto and switch statements.	4
9.	2	Execute programs to understand simple For loop and nested loops.	4
10.	2	Execute programs using While Loop and nested while loop.	2
11.		Execute programs using Do-while Loop and nested Do-while loop.	2
12.	3	i. Execute program to display 1-D and 2-D array. ii. Execute programs on arrays. (Sorting, finding particular value etc.)	4
	3	Execute a program for matrix addition.	2
13.		Execute a program for matrix multiplication	2
14.	3	Execute programs using String functions strlen(), strcpy, strcmp(), strcmp(), strchr(), strcat()	2
15.	4	Execute a program for math and other functions like sqrt(), pow(), ceil(), round(), sin(), cos(), tan(), div(), abs() etc	2
16.	4	Execute programs using functions and passing function arguments.	4
17.	4	Execute programs for pointer.	2
18.	4	Execute programs using recursive Functions.	2
19.	4	Execute program for call by reference	2
20.	5	Execute and execute programs with various features	2

Sr. No.	Unit No.	Practical Exercises (Outcomes in Psychomotor Domain)	Approx. Hrs. required
		of Structures	
21.	5	Execute program for structure using pointer	4
22.	5	Execute program for array of structure.	2
23.	5	Execute program for structure within structure	2
24.	5	Execute and execute programs using Union	2
25.	5	Execute and execute programs for creating memory for structure variable using <malloc.h>	2
Total			64

8. SUGGESTED STUDENT ACTIVITIES

LIST OF PROPOSED ACTIVITY:

1. Students will prepare file for the above mentioned Practical
2. Prepare presentation and deliver seminar on various topics covered like String functions, Pointers, Arrays, File Functions, Structures and Unions.
3. Students are expected to develop minimum one program of particular topic as an example to exhibit real life application.

9. SPECIAL INSTRUCTIONAL STRATEGIES

1. Hands on windows 7 or XP
2. Installation of C Compiler
3. Students should consider various types of problem definition for implementing programs.

