GOVERNMENT POLYTECHNIC, NAGPUR.

(An Autonomous Institute of Govt. of Maharashtra)

COURSE CURRICULUM

PROGRAMME	:	**DIPLOMA IN CM/IT**
LEVEL NAME	:	**PROFESSIONAL COURSES**
COURSE CODE	:	**CM401E**
COURSE TITLE	:	**PROGRAMMING IN 'C'**
PREREQUISITE	:	**NIL**
TEACHING SCHEME	:	**TH: 03; TU: 00; PR: 04(CLOCK HRs.)**
TOTAL CREDITS	:	**05(1 TH/TU CREDIT = 1 CLOCK HR., 1 PR CREDIT = 2 CLOCK HR.)**
TH. TEE	:	**03 HRs**
PR. TEE	:	**02 HRs (External)**
PT.	:	**01 HR**

❖ RATIONALE:

Engineering students should know the basics of programming in open source environment. This course describes the basics of problem solving and logic development. It also describes basics of programming using C programming language. C is the most commonly used structured programming language. It is very useful course for understanding the higher level knowledge in the field of software engineering and learning advanced object oriented languages.

❖ COURSE OUTCOMES:

After completing this course students will be able to–

1. Analyze the problem, design algorithm and draw flowchart.
2. Apply concept of modular programming.
3. Select and use appropriate branching and looping statements, functions, and data types available in C, as required
4. Develop a program in'C'..
5. Write and execute programs in C.
6. Debug and execute programs in C.

❖ COURSE DETAILS:

A. THEORY :

Units	Specific Learning Outcomes (Cognitive Domain)	Topics and subtopics	Hrs
1. Fundamentals of programming	1. Define various terms related to programming 2. Enlist C tokens, keywords, various rules, symbols, data types. 3. Identify valid / invalid variable names, syntax errors 4. Compare various terms. 5. Write an algorithm and draw flowchart for simple problems. 6. Describe structure of C program.	1.1 History of C, constants , Definition / description of program, syntax rules, character set, C tokens, keywords, identifiers, rules for identifiers, constants, types of constants, backslash character constants, variables, data types, primary, derived, user defined data types, size and range of data types, declaration and initialization of variables, defining symbolic constants, underflow and overflow of data. 1.2 Definition / description of algorithm, flowchart, High level language, Low level language , compiler, interpreter. 1.3 Flowchart symbols, algorithm and flowchart for simple problems 1.4 Structure of C program, various sections of C program and their purpose. 1.5 Process of creating, compiling and executing program. 1.6 Sample programs.	4
2. Basic I/O functions, Operators and Expressions in C	1. List various Formatted and unformatted I/O functions of C language along with syntax, its use and example. 2. List and describe various operators, their types and uses. 3. Write valid C expressions using various operators. 4. Identify and correct the errors in expressions and statements. 5. Write simple C programs.	**2.1** Syntax, use, examples of scanf(), printf(), getchar(), putchar(), gets(), puts() and examples, various format codes (%c, %d, %e, %f, %g, %h, %i, %o,%s, %u, %x), output format flags (-, +, 0, #), Rules for scanf(). 2.2 Definition and example of operands, operators, expression. 2.3 Symbols, uses and examples of various Arithmetic, Relational, logical, boolean , bitwise , assignment, ternary, increment (++) and decrement (--) operators, rules for ++ , -- operators supported by C. 2.4 Writing and evaluation of arithmetic expression. Precedence	6

		of arithmetic operators, rules for evaluation of expression, inbuilt math functions: sin(), cos(), tan(), sqrt(), pow(), log(), log10(), exp(). 2.5 Type casting: Implicit and Explicit, precedence and associatively of operators, their rules. 2.6 Simple C programs using various operators and typecasting.	
3. Decision Making, Branching and Looping	1.List and write Decision making statements, unconditional jump statement, loop statements supported by C. 2. Select and apply appropriate Decision making statements, unconditional jump statement, loop statements supported by C as per requirement. 3. Compare various loop statements, supported by C. 4. List rules for nesting loops. 5. Identify and correct the errors. 6. Write C programs using decision making and loop statements.	3.1 Syntax , working, flowchart and example of if , if- else, nested if- else , else- if ladder, switch statement, rules for switch statement, goto statement (forward jump and backward jump). 3.2 Syntax , working, flowchart, comparison and example of for loop, while loop and do while loop. 3.3 Jumping within loops and jumping out of loop. 3.4 Syntax , working and use of break, continue and exit(); 3.5 Algorithm, flowchart programs using if , if- else, nested if- else , else-if ladder, switch statement, goto , for loop , while loop, do - while loop, nested loops .	**10**
4. Functions	1. Define function and terms related to function. 2. Enlist and define type of functions, scope rules. 3. Enlist categories of of functions and state their examples. 4. State and describe ways of passing values to functions 5. Define and describe recursion. 6. Write programs based on functions.	4.1 Definition / description of modular programming, User Defined Function (UDF), function prototype, local, global parameters and formal, actual arguments, Library and user defined functions. 4.2 Elements of UDF, definition of UDF, return values and their types, function call, function declaration, 4.3 Function categories: function with no argument and no return value, with no argument but return value, with argument and no return value, with no argument and one return value, that returns multiple values. 4.4 Pass by value and by reference,	**8**

		rules for pass by pointers. 4.5 Algorithm, flowchart and programs based on functions, call by value, call by reference, recursion.	
5. Array and pointers	1. Define Array, string, pointer and their related terms. 2. Declare array and perform various operations on array. 3. Declare string, perform various operations on string and use various string library functions 4. Declare pointer variable, perform operations on pointer variables. 5. Enlist benefits of various data structures. 6. Write programs based on arrays, strings and pointers.	5.1 Definition of subscript, subscripted variable, derived data type, array, character array, string, NULL character, pointer 5.2 Declaration, compile time and run time initialization, accessing and processing elements of one dimensional array, 2D array. 5.3 Declaration, initialization of character array, using getchar(), gets(), scanf(), putchar(), puts(), printf() to input /output strings, Arithmetic operations on characters. 5.4 Syntax, working, use and examples of string handling functions like strlen(), strcpy(), strcmp(),strcat(). 5.5 Symbol and use of indirection operator, address operator. Initialization of pointer variable, legal arithmetic on pointers, comparison of Array and Pointer. 5.6 Benefits / advantages of using arrays, strings and pointers. 5.7 Programs based on arrays, strings, Pointers, Pointers and arrays.	**10**
6. Structure and Union	1. Define structure, union and their related terms. 2. Design, Define, declare structure, perform operations on individual members of structure and union. 3. Write programs based on Structure and Union	6.1 Description of user defined data type, Structure and union variables, member operator. 6.2 Structure definition, declaration and initialization. Accessing, copying, comparing structure members, size of structure, rules for initialization of structure, comparing structure and union. 6.3 Examples of array of structures, array within structure and structure within structure. 6.4 Benefits / advantages of using structures. 6.5 Comparison of array and structure. 6.6 Programs based on Structure,	**10**

		Union , array of Structures , array within structure and structure within structure .	
			Total Hrs. **48**

B. LIST OF PRACTICALS/LABORATORY EXPERIENCES/ASSIGNMENTS:

(Note: Perform all practicals in LINUX environment)

Practical	Specific Learning Outcomes (Psychomotor Domain)	Units	Hrs.
1.	Write and execute two simple C programs using standard formatted I/O functions. Write and execute four simple C programs using various operators and typecasting	2.Basic I/O , Operators and Expressions in C	04
2	Write and execute two programs each using if - else, if - elseif ladder.	3.Decision Making and Branching and Looping	02
3	Write and execute two programs each using nested if-else and switch statement.		04
4	Write and execute two programs using while loop		02
5	Write and execute two programs using do- while loop		02
6	Write and execute two programs each using for loop and nested for loop.		04
7	Write and execute 2 programs using function based on parameters passing by value.	4.Functions	02
8	Write and execute 2 programs using function based on parameters passing by reference.		04
9	Write and execute a programs using function based on recursion.		02
10	Write and execute two simple programs each using 1D and 2D arrays.	5.Array and pointers	04
11	Write and execute a simple program using character array.		02
12	Write and execute two simple programs each using string library functions.		02
13	Write and execute two simple programs each using pointers.		04
14	Write and execute a simple program using array of pointers.		02
15	Write and execute a simple program each using structure and union.	6.Structure and Union	02
16	Write and execute two simple programs using array of structures and array within structure.		04
17	Write and execute a program using structure within structure.		02
18	Mini project based on above concepts		12
		Skill Assessment	**04**
		Total Hrs	**64**

❖ SPECIFICATION TABLE FOR THEORY PAPER:

Unit No.	Units	Levels from Cognition Process Dimension			Total Marks
		R	U	A	
01	Fundamentals of programming	06(00)	04(04)	00(00)	10(04)
02	Basic I/O , Operators and Expressions in C	06(02)	04(04)	00(00)	10(06)
03	Decision Making, Branching and Looping	02(00)	08(04)	06(06)	16(10)
04	Functions	02(00)	08(00)	00(06)	10(06)
05	Arrays and Pointers	02(04)	06(04)	06(00)	14(08)
06	Structures and Unions	00(02)	04(04)	06(00)	10(06)
	Total	**18(08)**	**34(20)**	**18 (12)**	**70 (40)**

R – Remember U – Understand A – Analyze / Apply

❖ QUESTION PAPER PROFILE FOR THEORY PAPER:

Q. No	Bit 1			Bit 2			Bit 3			Bit 4			Bit 5			Bit 6			option
	T	L	M	T	L	M	T	L	M	T	L	M	T	L	M	T	L	M	
01	1	R	2	2	R	2	3	R	2	4	R	2	5	R	2	2	R	2	5/7
	6	R	2																
02	1	R	4	1	U	4	2	U	4	1	U	4	2	U	4				3/5
03	2	R	4	3	U	4	4	U	4	3	U	4	5	U	4				3/5
04	3	U	4	4	U	4	6	U	4	5	R	4	6	U	4				3/5
05	3	A	6	5	U	6	4	A	6										2/3
06	5	A	6	6	A	6	3	A	6										2/3

T= Unit/Topic Number L= Level of Question M= Marks

R-Remember U-Understand A-Analyze/ Apply

❖ ASSESSMENT AND EVALUATION SCHEME:

	What		To Whom	Frequency	Max Marks	Min Marks	Evidence Collected	Course Outcomes
Direct Assessment Theory	CA (Continuous Assessment)	Progressive Test (PT)	Students	Two PT (average of two tests will be computed)	20	--	Test Answer Sheets	1, 2, 3
		Assignments		Continuous	10	--	Assignment Book / Sheet	1, 2, 3
	TEE (Term End Examination)	End Exam	Students	End Of the Course	**70**	**28**	Theory Answer Sheets	1, 2, 3
				Total	**100**	**40**		
Direct Assessment Practical	CA (Continuous Assessment)	Skill Assessment	Students	Continuous	20	--	Rubrics & Assessment Sheets	4,5,6
		Journal Writing		Continuous	05	--	Journal	4,5,6
				TOTAL	**25**	**10**		
	TEE (Term End Examination)	End Exam	Students	End Of the Course	**50**	**20**	Rubrics & Practical Answer Sheets	4,5,6
Indirect Assessment	Student Feedback on course		Students	After First Progressive Test	Student Feedback Form			1, 2, 3, 4,5,6
	End Of Course			End Of The Course	Questionnaires			

❖ SCHEME OF PRACTICAL EVALUATION:

S.N.	Description	Max. Marks
1	Writing program and Drawing flow chart	20
2	Execution of program	20
3	Viva voce	10
	TOTAL	**50**

❖ MAPPING COURSE OUTCOMES WITH PROGRAM OUTCOMES:

1. Computer Engineering:-

Course Outcomes	Program Outcomes (POs)										PSOs	
	1	2	3	4	5	6	7	8	9	10	1	2
1	-	3	-	-	-	-	-	-	-	-	3	2
2	-	3	-	-	-	-	-	-	-	-	3	2
3	-	3	-	-	-	-	-	-	-	-	3	-
4	-	3	2	2	-	-	-	2	2	2	3	-
5	-	3	2	2	-	-	-	2	2	2	3	-
6	-	3	2	2	-	-	-	2	2	2	3	-

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

2. Information Technology:-

Course Outcomes	Program Outcomes (POs)										PSOs	
	1	2	3	4	5	6	7	8	9	10	1	2
1	-	3	-	-	-	-	-	-	-	-	-	3
2	-	3	-	-	-	-	-	-	-	-	-	3
3	-	3	-	-	-	-	-	-	-	-	-	3
4	-	3	2	2	-	-	-	2	2	2	-	3
5	-	3	2	2	-	-	-	2	2	2	-	3
6	-	3	2	2	-	-	-	2	2	2	-	3

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

❖ **REFERENCE & TEXT BOOKS:**

S.N.	Title	Author, Publisher, Edition and Year Of publication	ISBN Number
1.	Programming in ANSI 'C'	E. Balagurusamy , Publication Mc graw Hill Education , 6th Edition,2012.	13 : 9781259004612
2.	Let us C	Yashwant Kanetkar BPB Publication , 13th Edition,2015.	13: 9788183331630

❖ **E-REFERENCES:**

- **http://www.computer-pdf.com/programming/c-cpp/284-c-programming-book.html accessed on 30th March 2016**
- **http://www.w3schools.org.in accessed on 30th March 2016**
- **http://www.cquestions.com/ accessed on 21st April 2016**

❖ **LIST OF MAJOR EQUIPMENTS/INSTRUMENTS WITH SPECIFICATION**

1. Computer (Dual Core or above)
2. Network printer.
3. TC/GCC compiler

❖ **LIST OF EXPERTS & TEACHERS WHO CONTRIBUTED FOR THIS CURRICULUM:**

S.N.	Name	Designation	Institute / Industry
1.	Mr. S.P. Lambhade	Head of Computer Engineering	Government Polytechnic, Nagpur.
2.	Dr. Mrs. A.R. Mahajan	Head of Information Technology	Government Polytechnic, Nagpur.
3	Mrs.V.A. Raje	Lecturer in Computer Engineering	Government Polytechnic, Nagpur.
4	Ms. D.M. Shirke	Lecturer in Computer Engineering	Government Polytechnic, Nagpur.
5	Ms. G. B. Chavan	Lecturer in Computer Engineering	Government Polytechnic, Nagpur.
6	Mr. Atul Upadhyay	CEO	Vista Computers , Ram Nagar, Nagpur
7	Prof. N. V. Chaudhari	Asst. Professor (CSE)	DBACEO, Wanadongri, Nagpur
8	Prof. Manoj Jethawa	HOD Computer Science	Shri Datta Meghe Polytechnic, Nagpur

(Member Secretary PBOS) **(Chairman PBOS)**