

## Course Code: 104

### Course Title: Computer Programming & Programming Methodology (CPPM)

<b>Course Code</b>	104																																																						
<b>Course Title</b>	Computer Programming & Programming Methodology (CPPM)																																																						
<b>Credits</b>	4																																																						
<b>Course Category</b>	Major Course																																																						
<b>Level of Course</b>	200-299 ( Intermediate Level )																																																						
<b>Teaching per Week</b>	4 Hrs. ( 2 Hours Theory + 4 Hours Practical work )																																																						
<b>Minimum weeks per Semester</b>	15 (Including class work, examination, preparation etc.)																																																						
<b>Review / Revision</b>	2022-2023																																																						
<b>Implementation Year:</b>	A.Y. 2023-2024																																																						
<b>Purpose of Course</b>	<ul style="list-style-type: none"> <li>- Computer programming is a process that leads from an original formulation of a computing problem to executable computer programs.</li> <li>- Programming involves activities such as analysis, developing, understanding, generating algorithms, verification of requirements of algorithms including their correctness, and implementation (commonly referred to as coding) of algorithms in a target programming language.</li> <li>- To emphasis on concepts of Compiler based programming language, structure of code, algorithms, flow-charts, problem solving attitude, concepts of variables and declaration mechanism of different datatypes, simple I/O statements, conditional statements, loops, compound iterations, strings and certain inbuilt functions, header files, concepts of arrays and one dimensional numeric array operations, numeric inbuilt functions and concepts of pointers</li> </ul>																																																						
<b>Course Objective</b>	To introduce students the essentials of computer Programming and programming methodology using C Programming language.																																																						
<b>Pre-requisite</b>	-																																																						
<b>Course Outcomes</b>	<p>CO1: Students will be able to learn programming concept of compiler based programming language.</p> <p>CO2: Students will be proficient working on conditional statements, iterative Statements and fundamentals of programming concepts using C and Python.</p> <p>CO3: Students will be able to understand and implement conditional statements and improve their logical and reasoning abilities.</p> <p>CO4: Students will be able to develop understanding about iterative statements and their practical use.</p> <p>CO5: Students will learn about arrays and pointers.</p>																																																						
<b>Mapping between Course Outcomes(CO) with Program Specific Outcomes(PSO)</b>	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th></th> <th>PSO1</th> <th>PSO2</th> <th>PSO3</th> <th>PSO4</th> <th>PSO5</th> <th>PSO6</th> <th>PSO7</th> <th>PSO8</th> </tr> </thead> <tbody> <tr> <td>CO1</td> <td style="background-color: #cccccc;"></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>CO2</td> <td style="background-color: #cccccc;"></td> <td style="background-color: #cccccc;"></td> <td></td> <td></td> <td style="background-color: #cccccc;"></td> <td></td> <td></td> <td></td> </tr> <tr> <td>CO3</td> <td style="background-color: #cccccc;"></td> <td></td> <td></td> <td style="background-color: #cccccc;"></td> <td></td> <td></td> <td style="background-color: #cccccc;"></td> <td></td> </tr> <tr> <td>CO4</td> <td></td> <td></td> <td style="background-color: #cccccc;"></td> <td></td> <td></td> <td style="background-color: #cccccc;"></td> <td></td> <td style="background-color: #cccccc;"></td> </tr> <tr> <td>CO5</td> <td style="background-color: #cccccc;"></td> <td></td> <td></td> <td style="background-color: #cccccc;"></td> <td></td> <td style="background-color: #cccccc;"></td> <td style="background-color: #cccccc;"></td> <td style="background-color: #cccccc;"></td> </tr> </tbody> </table>		PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8	CO1									CO2									CO3									CO4									CO5								
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6	PSO7	PSO8																																															
CO1																																																							
CO2																																																							
CO3																																																							
CO4																																																							
CO5																																																							
<b>Course Content</b>	<p><b>UNIT-1: Introduction</b></p> <p>1.1 Concepts of Programming Language</p> <p style="padding-left: 20px;">1.1.1 Introduction of Source Code, Object Code and executable code</p> <p style="padding-left: 20px;">1.1.2 Algorithm and Flowchart</p> <p style="padding-left: 20px;">1.1.3 Concepts of Structured Programming Language</p> <p>1.2 Concepts of Editor, Interpreter and Compiler</p> <p style="padding-left: 20px;">1.2.1 Introduction of C program body structure</p> <p style="padding-left: 20px;">1.2.2 Character Set, concepts of variables and constants</p>																																																						

1.2.3 Identifiers, literals, Key words

1.2.4 Data types (signed and unsigned) (Numeric : int, short int, long, float, double) , (Character type: char, string) and void.

1.2.5 Concepts of source code, object code and executable code.

**UNIT-2: Input/Output Statements and Operators:**

2.1 Input/Output statements:

2.1.1 Concepts of Header files (STDIO, CONIO)

2.1.1.1 Concepts of pre-compiler directives.

2.1.1.2 Use of #include and #define

2.2 Input/Output Statements:

2.2.1 Input statements : scanf(), getc(), getch(), gets(), getchar()

2.2.2 Output Statements: printf(), putc(), puts(), putchar()

2.2.3 Type specifiers (formatting strings) : %d, %ld, %f, %c, %s, %lf

2.3 Operators :

2.3.1 Arithmetic operators ( +, -, \*, /, %, ++, --, )

2.3.2 Logical Operators ( &&, ||, ! )

2.3.3 Relational Operators ( >, =, <=, != )

2.3.4 Bit-wise operators ( &, |, ^, <> )

2.3.5 Assignment operators ( =, +=, -=, \*=, /=, %= )

2.3.6 Ternary Operator and use of sizeof() function.

2.4 Important Built-in functions:

2.4.1 Use of : ( strlen, strcmp, strcpy, strcat, strrev)

2.4.2 Use of : (abs(), floor(), round(), ceil(), sqrt(), exp(), log(), sin(), cos(), tan(), pow() and trunc())

**UNIT-3: Decision Making statements :**

3.1 if statements :

3.1.1 simple if statements

3.1.2 if...else statements

3.1.3 if...else if...else statements

3.1.4 Nested if statements.

3.2 Switch..case statements

3.2.1 Use of break and default

3.2.2 Difference between switch and if statements.

**UNIT-4: Iterative statements :**

4.1 Use of goto statement for iteration

4.2 while loop

4.3 do..while loop

4.4 for loop

4.5 Nested while, do..while and for loops

4.6 Jumping statement: (break and continue)

**UNIT-5: Concepts of Arrays and pointer**

5.1 Concepts of Single-dimensional Array

5.1.1 Numeric single dimensional Array

5.1.2 Numeric single dimensional array operations:

5.1.2.1 Sorting array in ascending or descending. (Bubble and selection)

5.1.2.2 Searching element from array (Linear Search)

5.1.3 Character Single dimensional Array

5.1.3.1 Character Single dimensional array operations:

5.1.3.2 Use of \0, \n and \t

5.2 Pointers:

5.2.1 Concepts of Pointers

5.2.2 Declaring and initializing int, float, char and void pointers

5.2.3 Pointer to single dimensional numeric array.

<b>Reference Books</b>	<ol style="list-style-type: none"> <li>1. Programming in C, Balaguruswami – TMH</li> <li>2. C: How to Program, Deitel &amp; Deitel - PHI</li> <li>3. C Programming Language, Kernigham &amp; Ritchie - TMH</li> <li>4. Programming in C, Stephan Kochan - CBS</li> <li>5. Mastering Turbo C, Kelly &amp; Bootle - BPB</li> <li>6. C Language Programming – Byron Gottfried - TMH</li> <li>7. Let us C, Yashwant Kanetkar - BPB Publication</li> <li>8. Magnifying C, Arpita Gopal - PHI</li> <li>9. Problem Solving with C, Somashekara - PHI</li> <li>10. Programming in C, Pradip Dey &amp; Manas Ghosh – Oxford</li> </ol>
<b>Teaching Methodology</b>	Class Work, Discussion, Lab work, Self-Study, Seminars and/or Assignments
<b>Evaluation Method</b>	<p>50% Internal assessment.</p> <p>50% External assessment..</p>