Programming and Problem Solving (23OMC103)

Overview of the Course

Course Basics

• Programme:	Master of Computer Applications
• Semester:	1
• Course Title:	Introduction to Programming and Problem Solving
• Course Code:	OMC103
• Course Credits:	3
• Course Type:	Core Theory Course

Course Outcomes (COs)

After the successful completion of this course, the student will be able to:

CO-1.Describe the fundamental concepts of computational thinking and problem-solving strategies. [L-1]

CO-2.Demonstrate the use of arrays, strings, structures, and unions in the 'C' programming language. [L-3]

CO-3.Demonstrate the use of re-usable code using functions in 'C'. [L-3]

CO-4.Describe and implement file handling mechanism in 'C' programs. [L-3]

Unit 1

Title: Computational Thinking

- What is computational thinking?
- Computational thinking approaches
- Information and Data – Converting Information to Data
- Data Types and Encoding

Unit 2

Title: Problem Solving and Programming Approaches

- Problem Solving Techniques
- Algorithms
- Flowcharts
- Pseudocode
- Classification and Characteristics of Programming Language
- Programming Paradigms

Unit 3

Title: Introduction to C Programming, Variables, and Constants

- Characteristics of ‘C’
- Structure of C Program
- The life cycle of the C Program
- First C Program
- Commands to run a C Program
- Comments Style in ‘C’
- Programming errors, Syntax errors, and semantic errors
- Logical and runtime errors

Unit 4

Title: Operators and Expressions in C, Input/Output (I/O) Functions

- Assignment Operators, Arithmetic Operators
- Relational Operators, Logical Operators
- Increment and Decrement Operators
- Conditional Operators
- Typecast Operators, sizeof Operator
- Associativity and precedence of operators
- Evaluation of Expressions

Unit 5

Title: : I/O Functions

- Types of I/O Functions
- Unformatted I/O Functions
- getchar(), putchar()
- gets(), puts(), getch(), putch()
- Formatted I/O functions
- Format specifiers
- scanf(), printf()

Unit 6

Title: Function

- Library functions
- Function declaration and definition, Function prototype and call, Return Statement
- Function with and without arguments and return value
- Function call by value and call by reference
- Advantages of functions, Function call stack and activation records
- Recursive functions, Recursive Vs Iterations, Examples of recursive functions
- Static and Dynamic Linking

Unit 7

Title: Pointers

- Pointers and their characteristics
- Pointer declaration and assignment
- Dereferencing pointer variables
- Pointer arithmetic
- Pointers and functions
- Dynamic memory allocation – malloc(), calloc() realloc(), free() functions
- Memory leak and segmentation fault

Unit 8

Title: Arrays and Strings

Arrays -

- Single-dimensional array, Array declaration, Accessing elements of an array
- Initialization, Array operations (insert, delete, sort, and search)
- Two-dimensional arrays: Declaration and Initialization
- Operations on Matrices (addition, product, transpose)

Strings -

- Declaration and initialization of strings
- Input and Output of strings
- Formatting Strings

Unit 9

Title: Structures and Union

Structures -

- Need of structures, Declaring and defining a structure, Initialization of structure variables
- Accessing structure members, assignment of structure variables
- Size of a structure, Array of structures, Structure with arrays
- Nested structure, Structures and functions, Structures and Pointers, Self-referential structure

Union –

- Declaring and defining a union, Initialization and access of union variables, Size of a union, Nested unions

Unit 10

Title: File Handling

- Types of files
- File modes, Opening, closing, and end of a file.
- Character I/O functions - fputc() , fgetc()
- Integer I/O functions - putw(), getw()
- String I/O functions - fputs(), fgets()
- Formatted I/O functions - fprintf(), fscanf()
- Block Read/Write functions - fwrite(), fread()
- Random access to a file - fseek(), ftell(), rewind()
- Error handling in files

Course Resources

a. Essential Reading

1. David D. Riley and Kenny A. Hunt, (2014), Computational Thinking for the Modern Problem Solver, Chapman & Hall/CRC.
2. Yashavant Kanetkar, (2016), Let Us C, 14th Edition, BPB Publication.

b. Recommended Reading

1. E. Balagurusamy, (2015), Programming in ANSI C, 6th Edition, McGraw-Hill.
2. Brian W Kernighan & Dennis M Ritchie, (1988), The C Programming Language, 2nd Edition, Prentice Hall.
3. Steve Oualline, (2011), Practical C Programming, 3rd Edition, Orielly Publishers.

END