L	T	P	C
2	1	0	3

Course Code: CSE106
Semester: I

FUNDAMENTALS OF COMPUTER SCIENCE

Course Objectives:

This course will help the learner to formulate simple algorithms for arithmetic, logical problems. Translate the algorithms into ANSI C programs using operators, control flow statements, arrays, functions, pointers and structures.

UNIT - I **10 Periods**

General problem Solving concepts: Algorithm, and Flowchart for problem solving with Sequential Logic Structure, Decisions and Loops.

Introduction: Imperative languages - syntax and constructs of a specific language (ANSI C) - Types Operator and Expressions with discussion of variable naming and Hungarian Notation - Variable Names - Data Type and Sizes (Little Endian Big Endian) - Constants - Declarations - Arithmetic Operators - Relational Operators - Logical Operators - Type Conversion - Increment Decrement Operators - Bitwise Operators - Assignment Operators and Expressions - Precedence and Order of Evaluation - proper variable naming and Hungarian Notation

UNIT - II **10 Periods**

Control Flow Statements: Control Flow with discussion on structured and unstructured programming - Statements and Blocks - If-Else-If - Switch - Loops - while, do, for, break and continue - Goto Labels - structured and un structured programming.

Functions: Functions and Program Structure with discussion on standard library - Basics of functions - parameter passing and returning type - C main return as integer, External, Auto, Local, Static, Register Variables - Scope Rules - Block structure - Initialization - Recursion - Preprocessor - Standard Library Functions and return types.

UNIT - III **13 Periods**

Pointers and Arrays: Pointers and address - Pointers and Function Arguments - Pointers and Arrays - Address Arithmetic - character Pointers and Functions - Pointer Arrays - Pointer to Pointer - Multi-dimensional array and Row/column major formats - Initialization of Pointer Arrays - Command line arguments - Pointer to functions - complicated declarations and how they are evaluated.

Structures: Basic Structures - Structures and Functions - Array of structures - pointer of structures - Self-referral Structures - Table look up - Typedef - Unions - Bit-fields

UNIT - IV **12 Periods**

Files Input Ouput: Input and Output Standard I/O - Formatted Output -printf - Formatted Input - scanf - Variable length argument list - file access including FILE structure - fopen - stdin, sdtout and stderr - Error Handling including exit - perror and error.h - Line I/O - related miscellaneous functions.

UNIX system Interface: Unix system Interface - File Descriptor - Low level I/O - read and write, Open, create, close and unlink - Random access -lseek - Discussions on Listing Directory - Storage allocator.
Programming Method: Debugging - Macro - User Defined Header - User Defined Library Function - makefile utility

TEXTBOOKS

1. B. W. Kernighan and D. M. Ritchi. "*The C Programming Language"*, PHI, Second edition,2015.
2. B. Gottfried. "*Programming in C"*, Schaum Outline Series, Third Edition, 2017.

REFERENCES

1. Herbert Schildt. "*C: The Complete Reference"*, McGraw Hill, Fourth Edition, 2017.
2. Yashavant Kanetkar. "*Let Us C"*, BPB Publications, Sixteenth Edition, 2017.

.

UNITWISE LEARNING OUTCOMES

Upon successful completion of each unit, the learner will be able to

Unit I	• Identify the basic concepts of problem solving techniques • Develop programs to perform different mathematical operations using operators
Unit II	• Select an appropriate construct to solve the given problem • Design programs using functions, branching and looping constructs
Unit III	• Illustrate the operations on arrays • Experiment with structures and pointers for developing a given application
Unit IV	• Construct simple application using formatted input and output statements • Demonstrate the file access programs using built in functions

COURSE LEARNING OUTCOMES

Upon successful completion of this course, the learner will be able to

- Illustrate a functional hierarchical code organization in ANSI C
- Interpret the different operators for performing mathematical operations
- Construct the programs using functions, branching and looping
- Demonstrate pointers and arrays in a given application
- Build an application to perform file access using built in function
- Propose a solution for the real time problems using C constructs