|  |  |  |
| --- | --- | --- |
|  | **PROGRAMMING IN C LTPC** | **LPTC** |

**3003**

**COURSE OBJECTIVES:**

• To understand the constructs of C Language.  
• To develop C Programs using basic programming constructs  
• To develop C programs using arrays and strings  
• To develop modular applications in C using functions  
• To develop applications in C using pointers and structures  
• To do input/output and file handling in C

|  |  |  |
| --- | --- | --- |
| **UNIT I** | **BASICS OF C PROGRAMMING 9** | **9** |

Introduction to programming paradigms – Applications of C Language - Structure of C program - C programming: Data Types - Constants – Enumeration Constants - Keywords – Operators: Precedence and Associativity - Expressions - Input/Output statements, Assignment statements – Decision making statements - Switch statement - Looping statements – Preprocessor directives - Compilation process

|  |  |  |
| --- | --- | --- |
| **UNIT II** | **ARRAYS AND STRINGS 9** | **9** |

Introduction to Arrays: Declaration, Initialization – One dimensional array – Two dimensional arrays - String operations: length, compare, concatenate, copy – Selection sort, linear and binary search.

|  |  |  |
| --- | --- | --- |
| **UNITIII** | **FUNCTIONS AND POINTERS 9** | **9** |

Modular programming - Function prototype, function definition, function call, Built-in functions (string functions, math functions) – Recursion, Binary Search using recursive functions – Pointers – Pointer operators – Pointer arithmetic – Arrays and pointers – Array of pointers – Parameter passing: Pass by value, Pass by reference.

|  |  |  |
| --- | --- | --- |
| **UNITIV** | **STRUCTURES AND UNION 9** | **9** |

Structure - Nested structures – Pointer and Structures – Array of structures – Self referential structures – Dynamic memory allocation - Singly linked list – typedef – Union - Storage classes and Visibility.

|  |  |  |
| --- | --- | --- |
| **UNIT V** | **FILE PROCESSING 9** | **9** |

Files – Types of file processing: Sequential access, Random access – Sequential access file - Random access file - Command line arguments.

**COURSE OUTCOMES:**

**Upon completion of the course, the students will be able to**

CO1: Demonstrate knowledge on C Programming constructs  
CO2: Develop simple applications in C using basic constructs  
CO3: Design and implement applications using arrays and strings  
CO4: Develop and implement modular applications in C using functions.  
CO5: Develop applications in C using structures and pointers.  
CO6: Design applications using sequential and random access file processing.

**TOTAL : 45 PERIODS**

**TEXT BOOKS:**

1. ReemaThareja, “Programming in C”, Oxford University Press, Second Edition, 2016.  
2. Kernighan, B.W and Ritchie,D.M, “The C Programming language”, Second Edition, Pearson Education, 2015.

**REFERENCES:**

1. Paul Deitel and Harvey Deitel, “C How to Program with an Introduction to C++”, Eighth edition, Pearson Education, 2018.  
2. Yashwant Kanetkar, Let us C, 17th Edition, BPB Publications, 2020.  
3. Byron S. Gottfried, “Schaum’s Outline of Theory and Problems of Programming with C”, McGraw-Hill Education, 1996.  
4. Pradip Dey, Manas Ghosh, “Computer Fundamentals and Programming in C”, Second Edition, Oxford University Press, 2013.  
5. Anita Goel and Ajay Mittal, “Computer Fundamentals and Programming in C”, 1st Edition, Pearson Education, 2013.

|  |  |  |
| --- | --- | --- |
|  | **PROGRAMMING IN C LABORATORY** | **LTPC** |

**COURSE OBJECTIVES:**

• To familiarise with C programming constructs.  
• To develop programs in C using basic constructs.  
• To develop programs in C using arrays.  
• To develop applications in C using strings, pointers, functions.  
• To develop applications in C using structures.  
• To develop applications in C using file processing.

**LIST OF EXPERIMENTS:**

1. I/O statements, operators, expressions  
2. decision-making constructs: if-else, goto, switch-case, break-continue  
3. Loops: for, while, do-while  
4. Arrays: 1D and 2D, Multi-dimensional arrays, traversal  
5. Strings: operations  
6. Functions: call, return, passing parameters by (value, reference), passing arrays to function.  
7. Recursion  
8. Pointers: Pointers to functions, Arrays,Strings, Pointers to Pointers, Array of Pointers  
9. Structures: Nested Structures, Pointers to Structures, Arrays of Structures and Unions.  
10. Files: reading and writing, File pointers, file operations, random access, processor directives.

**TOTAL : 60 PERIODS**

**COURSE OUTCOMES:**

CO1: Demonstrate knowledge on C programming constructs.  
CO2: Develop programs in C using basic constructs.  
CO3: Develop programs in C using arrays.  
CO4: Develop applications in C using strings, pointers, functions.  
CO5: Develop applications in C using structures.  
CO6: Develop applications in C using file processing.