|  |  |  |  |
| --- | --- | --- | --- |
| **Course Code** | **Course Title** | **Credits** | **Year of Introduction** |
| **ECE 1103** | **Computer Programming** | 3.00 | 1st Year odd semester |

|  |  |
| --- | --- |
| **Course Type** | Compulsory |
| **Contact Hours** | 3 hours per week |
| **Prerequisite** | None |
| **Lecture Room** | Respective allocated class room |
| **Teaching methods** | Lecture, ppt slide |

|  |  |
| --- | --- |
| **Course Description** | This hands-on C programming course provides a comprehensive introduction to the ANSI C language, emphasizing portability and structured design. Students are introduced to all major language elements including fundamental data types, flow control, and standard function libraries. Thorough treatment is given to the topics of string and character manipulation, dynamic memory allocation, standard I/O, macro definition, and the C runtime library. The course explains the use of aggregate structures, unions, and pointers early on so the students can practice extensively in the hands-on labs. Structured programming constructs and various functions are also covered. Emphasis is given to the processing of command line arguments and environment variables so students will be able to write flexible, user-friendly programs. The course also includes coverage of portability tips drawn from experienced programmers working in production environments. Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency. |
| **Course Content** | **Introduction to computer programming:** Algorithm, Writing, debugging and running programs using C/C++ compiler.  **C/C++ Basics:** Different Data types and their range, Operator and operands and its precedence, input/output, conditional operators, loops nested structure, error handling, built-in functions.  **Functions and Arrays:** Writing & calling of User – defined functions, Recursive functions, scope of variables, introduction to one-dimensional arrays, multi-dimensional arrays and array manipulation.  **Pointers and Strings:** Introduction to pointers, pointers and array, pointers and functions, String I/O, String-based built-in functions, String operations, pointer and string.  **Files:** Introduction to files in C/C++, opening, closing and updating binary and sequential files.  **Advanced topics:** Operations on bits, register variable, Pre-processors and graphics in C/C++ |
| **Course Objectives** | The Objectives of the course is given below:   1. To provide concepts of Computer Programming to instruct computers according to user syntaxes. 2. To make students understands about the inner working mechanisms of compiler and interpreter. 3. To provide exposure to problem -solving through programming 4. To train the student to the basic concepts of the C -programming language. 5. To make students to be capable in solving real life problems using C language. 6. Application of C language in real life system design. |
| **Course Outcomes (COs)** | Followings are outcomes of the course:  **CO1:** **Clarify** the basics of computer programming with C.  **CO2: Apply** decision making statements, iterative statements.  **CO3: Analyze** the concept of variables and different data types.  **CO4:** **Design** new programs using theconcept of modular programming in C. |
| **Text Book** | 1. Programming with ANSI C, Tata McGraw Hill by E. Balagurusamy. 2. Teach Yourself C by Herbert Schildt |
| **Reference Readings** | 1. The C Programming Language by Kernighan and Ritchie 2. The Art of Computer Programming by D.E. Knuth. 3. C programming tutorial at tutorialpoints.com 4. Geeks for Geeks tutorials. |
| **Teaching Strategies** | Lectures, Field practice (group work), Industrial training, Assignments, Tutorials, Report writing, Experiential learning, Cooperative learning, Collaborative learning, Projects, Role play, Presentations, Quizzes, Self-study |
| **Assignments Strategies** | Besides regular homework, students are given additional tasks to complete and practice. Based on those tasks, additional assignment-based test is taken to count as assignment mark. |
| **Assessment Strategies** | The total marks are distributed for the examinations as shown.   1. Short Question/Definition type tests 2. Explaining briefly on basic concepts 3. Detailed Testing on topics through long answers/problem solving to understand in depth knowledge 4. To test the application capabilities based on concepts taught in the class |

**Student evaluation**

|  |  |  |
| --- | --- | --- |
| 1. | Class performance | 08 % |
| 2. | Class test | 20 % |
| 3. | Final Examination | 72 % |

**Mapping of Course Outcomes (COs) with Program Outcomes (POs) and Bloom’s Taxonomy:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| POs COs | | **CO1** | **CO2** | **CO3** | **CO4** |
| **Program Outcomes** | **PO1** | √ |  | √ |  |
| **PO2** |  | √ |  |  |
| **PO3** |  |  |  | √ |
| **PO4** |  |  |  |  |
| **PO5** |  |  |  |  |
| **PO6** |  |  |  |  |
| **PO7** |  |  |  |  |
| **PO8** |  |  |  |  |
| **PO9** |  |  |  |  |
| **PO10** |  |  |  |  |
| **PO11** |  |  |  |  |
| **PO12** |  |  |  |  |
| **Domain of Bloom’s Taxonomy** | **Cognitive or**  **Knowledge** | √ |  | √ |  |
| **Affective or Attitude** |  |  |  | √ |
| **Psychomotor or Skill** |  | √ |  |  |

**Lecture Schedule:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week 1** | | **Introductions** | CT 1 | | |
| Class 1 | | * Introductory lessons, Basic Concepts of compiler, interpreter, algorithm and flowchart. * Write C code, compile and execution of code, C code debugging |
| Class 2 | | * Program structure, variable, operators and data types. |
| Class 3 | | * Format specifier, Valid variable declaration, identifier, token. * Data types in C, built in data type and user defined data type |
| **Week 2** | | **Control Statement** |
| Class 4 | | * Decision making statements with if, if\_else, if\_else block and switch |
| Class 5 | | * Looping structures with for, while, do-while. |
| Class 6 | | * Different of while and do\_while loop. |
| **Week 3** | | **Control statement** |
| Class 7 | | * Counter based and sentinel based looping. |
| Class 8 | | * Application of looping. |
| Class 9 | | * Coding using looping, time requirement analysis of different looping. * Selecting loop in contest programming which solves time and space limitation errors. |
| **Week 4** | | **Array, Sorting and Searching** | CT 2 | | |
| Class 10 | | * Sequential variable declaration, Array processing |
| Class 11 | | * Declare array and different types (int, float, double) of array, indexing of array. * Scanning array elements using loop, Solve the anomaly of scanning integer variable after scanning character variable. |
| Class 12 | | * Different array operation  1. Scan 2. Traverse 3. Insert element 4. Delete element |
| **Week 5** | | **Array, Sorting and Searching** |
| Class 13 | | * Searching and sorting array, Bubble sort, Quick sort. |
| Class 14 | | * Linear search, Binary search. |
| Class 15 | | * Copying one array into another array, Reversing array, Memory concept behind array. * Review all previous topics. |
| **Week 6** | | **Functions** |
| Class 16 | | * Definition of function, Necessity of function. * Built in library function, User defined function, Function prototype, Function body. |
| Class 17 | | * Function return type, parameter, function local variable, static variable in function. * Function calling, caller and called function, control flow of code after function calling. |
| Class 18 | | * Call by value and call by reference. * Array passing as function parameter. * Return array to caller function. |
| **Week 7** | **String** | | | CT 3 |
| Class 19 | * String processing in C using character array. | | |
| Class 20 | * Scan complete array using %s and gets() function. | | |
| Class 21 | * Length of string, comparing two string, copy one string into another, reversing string. | | |
| **Week 8** | **String** | | |
| Class 22 | * Use of some built in string function  1. strlen() 2. strcat() 3. strcpy() 4. strcmp() 5. strstr() | | |
| Class 23 | * Application of string in different aspect of real life application. | | |
| Class 24 | * Review of string related topics. | | |
| **Week 9** | **User defined data types** | | |
| Class 25 | * User defined data type, Structures, Union, Enumeration. | | |
| Class 26 | * Defining structure, member variable, structure object, use of dot operator. | | |
| Class 27 | * Array of structure object, accessing a structure members. * Search and traverse in a structure array. * Union processing and concept of largest variable in union and application of union. | | |
| **Week 10** | **Pointers** | | | CT 4 |
| Class 28 | * What is pointer, declare pointer using asterisk (\*) operator, pointer to int, float, char and double type of data. * Pointer to structure and union, access members using arrow (->) operator. | | |
| Class 29 | * Application concept of pointer to store non-linear types of data. * Pointer of pointer, Passing pointer as function parameter, pointer as return type of function. | | |
| Class 30 | * Change of value using static variable in user defined function. * Review of the topics for the next class test. | | |
| **Week 11** | **File Handling** | | |
| Class 31 | * File Handling, preprocessor with define, include, macro, typedef. | | |
| Class 32 | * Read data and print data into a file. * Handling of different types of file like .txt .dat .c etc. | | |
| Class 33 | * Printing the current code program into console using file processing. * Application of file handling to store data. | | |
| **Week 12** | **Dynamic Memory Allocation** | | |
| Class 34 | * Dynamic memory allocation for variable. | | |
| Class 35 | * malloc() and calloc() function to allocate memory dynamically. | | |
| Class 36 | * Different of malloc() and calloc() function. * Case study of using of the malloc() and calloc() function. | | |
| **Week 13** | **Review on the total course** | | |  |
| Class 37 | * Review on Introduction and control statement. | | |
| Class 38 | * Review on Array, Sorting, Searching and Functions. | | |
| Class 39 | * Review on String, User defined data types, Pointers, File Handling and Dynamic memory allocation. | | |