

Prerequisite: There is no prerequisite in this course; however, students having any prior experience of programming are desirable.

Objective:

1. To provide exposure to problem-solving through programming.
2. To provide students with understanding of code organization and functional hierarchical decomposition with using complex data types.

Learning Outcomes:

Course Outcome	Description
CO1	Makes students gain a broad perspective about the uses of computers in engineering industry.
CO2	Develops basic understanding of computers, the concept of algorithm and algorithmic thinking.
CO3	Develops the ability to analyze a problem, develop an algorithm to solve it.
CO4	Develops the use of the C programming language to implement various algorithms, and develops the basic concepts and terminology of programming in general.
CO5	Introduces the more advanced features of the C language

Course Content:

Unit-1: Introduction to Programming: Basic computer organization, operating system, editor, compiler, interpreter, loader, linker, program development. Variable naming, basic function naming, indentation, usage and significance of comments for readability and program maintainability. Types of errors, debugging, tracing/stepwise execution of program, watching variables values in memory. Constants, Variables and data Types Character Set, C tokens, Keywords and Identifiers, Constants, Variables, Data types, Declaration of Variables, assigning values to variables, typedef, and Defining symbolic constants. printf & scanf function.

Unit-2: Operators and Expression: Introduction, Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operators, Increment and Decrement Operators, Conditional Operators, Special Operators, Evaluation of expressions, Precedence of arithmetic operators, Type conversions in expressions, Operator precedence and associativity.

Management Input and Output Operators: Introduction, reading a character, writing a character, formatted input, formatted output.

Unit-3: Decision Making Branching: Introduction, Decision making with IF statement, the IF-ELSE statement, nesting of IF-ELSE statement, ELSE-IF ladder, SWITCH statement, ternary operator, and the GOTO statement.

Looping: Introduction, the WHILE statement, the DO statement, The FOR statement, Break and Continue.

Unit-4: Array: Introduction, One-dimensional arrays, Two-dimensional arrays, arrays, Concept of Multidimensional arrays.

Handling of Character strings: Introduction, Declaring and initializing string variables, reading string from terminal, writing string to screen, String, Operations: String Copy, String Compare, String Concatenation and String Length (using predefined functions & without using them), Table of strings.

Unit-5: User-Defined Functions (UDF): Introduction, need for user-defined functions, the form of C function, elements of UDF, return values and their types, Calling a function, category of functions, Nesting of functions, Recursion, Functions with arrays, The scope and Lifetime of variables in functions, multi file program.

Structures and Unions: Introduction, Structure definition, declaring and initializing Structure variables, accessing Structure members, Copying & Comparison of structures, Arrays of structures, Arrays within structures, Structures within Structures, Structures and functions, Unions.

Unit-6: Pointers: Introduction, understanding pointers, Accessing the address of variable, Declaring and initializing pointers, accessing a variable through its pointer, Pointer expressions, Pointer increments and scale factor, Pointers and arrays, Pointers & character strings, Pointers & Functions, Function returning multiple values, Pointers and structures.

File Management in C and CONSOLE I/O: Introduction, Defining files and its Operations, Error handling during I/O operations, Random access files, Command line arguments. Types of files, File vs. Console, File structure, File attributes, Standard i/o, Formatted i/o, Sample programs.

Teaching Methodology:

This course is introduced to help students understand the discipline of programming. The programming language used to teach this course is C. Starting from the basic computer architecture, the student will slowly be exposed to program designing and later to programming fundamentals. The entire course is broken down into six separate units, from fundamentals of programming to some complex programming structures like pointers. This theory course is well complemented by a laboratory course under the name Software Development Fundamentals Lab in the same semester that helps a student learn with hand-on experience.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Based on Unit-1 & Unit-2
Test-2	25 Marks	Based on Unit-3 & Unit-4 and around 20-30% from coverage till Test-1
Test-3	35 Marks	Based on Unit-5 to Unit-6 and around 30% from coverage till Test-2
Assignment	10 Marks	

Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	100 Marks	

Learning Resources:

Tutorials and lecture slides on Software Development Fundamentals (will be added from time to time): Digital copy will be available on the JUET server.

Text Book:

- [1] Programming in ANSI C by E. Balguruswamy, Tata Mc-Graw Hill.
- [2] Programming With C, Schaum Series.

Reference Books/Material:

- [1] The 'C' programming language by Kernighan and Ritchie, Prentice Hall
- [2] Computer Programming in 'C' by V. Rajaraman, Prentice Hall
- [3] Programming and Problem Solving by M. Sprankle, Pearson Education
- [4] How to solve it by Computer by R.G. Dromey, Pearson Education

Web References:

- [1] <http://www2.its.strath.ac.uk/courses/c/>
Notes on C programming by University of Strathclyde Computer Centre. This tutorial was awarded the NetGuide Gold Award during the 1990s.

- [2] http://www.princeton.edu/~achaney/tmve/wiki100k/docs/C_%28programming_language%29.html

This site contains notes on C programming from Princeton University, USA.

These are very useful for students who are learning C as their first programming Language.

- [3] <http://www.stat.cmu.edu/~hseltman/Computer.html>

Online reference material on Computers and Programming from Carnegie Mellon University, Pittsburgh, USA

- [4] <http://projecteuler.net/>

Collection of mathematical problems which make you use your programming skills