IT 213: Structured Programming

Module Objectives

This module aims to introduce students to the discipline of computing with a focus on good program design, programming styles, and structured program development using a high-level programming language. The students shall also be introduced to the basic concepts in procedural abstraction, structured programming and top-down design with stepwise refinement. Laboratory work is essential in this module.

Contents:

Introduction to the C Language, Data Types and Variables, Input/Output Management, Expressions and operators, Selection statements, Loops, Arrays, Modular Programming with Functions, Pointers and Strings, Structures and Dynamic Memory Allocation, The Preprocessor, and File Input/Output

Detailed Course

Introduction to the C Language [3hrs]

- The C Language and its Advantages
- The Structure of a C Program, Writing C Programs, Debugging a C Program
- Examining and Running a C Application Program

Data Types and Variables [3hrs]

- Data Types (integer, floating, character, type conversion, type definitions, sizeof operator)

Input/Output Management [3hrs]

- Input/Output Management: printf(), scanf()
- Conversion specifiers
- Escape sequences

Expressions and operators [5hrs]

- Arithmetic operators: operator precedence and associativity
- Assignment operators
- Increment and decrement operators
- Expression evaluation
- Expression statements
- Relational operator, logical operator, arithmetic assignment,

Selection statements **[4hrs]**

- Logical expressions
- If statement
- Switch statement

Loops **[4hrs]**

- While
- Do..while
- For
- Exiting from a loop (break, continue, goto)

Arrays **[3hrs]**

- One-dimensional array
- Multidimensional array

Modular Programming with Functions **[4hrs]**

- Defining and calling functions
- Function declarations
- Arguments
- Return statement
- Program termination (exit function)

Pointers and Strings **[4hrs]**

- Pointer variables
- The address and indirection operators
- Pointer assignment
- Pointer as argument
- Pointer as return values
- Pointer Arithmetic
- Using pointers for array processing
- String literals, variables
- Reading and writing strings
- Using the C string library: strcpy, strlen, strcat, strcmp
- Arrays of strings

Structures and Dynamic Memory Allocation **[4hrs]**

- Structures, Arrays of Structures
- Passing Structures to Functions
- Nested Structures

- Ø Unions
- Ø enumerations
- Ø Dynamic Memory Allocation (malloc, calloc, realloc)
- Ø Deallocating storage
- Ø Linked list (-> operator, creating, displaying, searching)

The Preprocessor [4hrs]

- Ø How the preprocessor works
- Ø The C Preprocessor and the #include and #define Directives
- Ø Macro definitions (simple, parameterized macros), general properties of macros
- Ø #if and #endif directives
- Ø The defined operator

File Input/Output [4hrs]

- Ø Streams: file pointers, standard streams and redirection, text files versus binary files
- Ø File operations: opening a file, modes, closing a file, attaching a file to an open stream, obtaining file names from the command line
- Ø File Input and Output, reading/writing data, structure to files, random access

Text Book:

K. N. King, K.N. King, C Programming: A Modern Approach, W W Norton & Co Inc (February 1996)

Brian W. Kernighan, Dennis M. Ritchie ,C Programming Language, 2nd Edition Prentice Hall; 2 edition (April 1, 1988)

Reference book:

- David Griffiths, Dawn Griffiths**, Head First C**, O'Reilly Media; 1 edition (April 19, 2012)

- Clovis L. Tondo (Author), Scott E. Gimpel (Author), "The C Answer Book: Solutions to the Exercises in 'The C Programming Language,' Second Edition", Prentice Hall; 2nd edition (November 11, 1988)

- Peter van der Linden, Expert C Programming: Deep C Secrets, Prentice Hall; 1st edition (June 24, 1994)