PROGRAMMING METHODOLOGY

1. Introduction

Objective: This course teaches the basics of solving problems by writing computer programs. At the end of the course, a student will be able to think programmatically and write programs in a procedural/ imperative language. The primary language used for programming is 'C.'

Credits: 3

2. Course Outline

UNIT – I
Introduction to problem solving; Problems and problem instances; Informal approach to program design: generalisation, special cases, algorithms, breaking down a problem into functions, input and output.

UNIT – II
Introduction to the 'C' programming language; program structure; main() function; unnamed and named blocks; basic data types, variables, declaration and definition; initialisation and assignment; arithmetic operators and precedence; implicit and explicit type conversions; arrays; boolean variables and logical operators.

UNIT – III
Control structures: branching and iteration; functions and parameters; break(), return() and exit() functions; local and global variables; function prototypes.

UNIT – IV
Pointer variables and dynamic structures; static and dynamic (run-time) memory structures; static variables; breaking a program across multiple files; creating and linking libraries.

UNIT – V
Detecting and correcting common errors; debugging and debuggers; documenting programs; good programming practices; programming exercise (writing a program of at least 200 lines split across multiple files).

3. Reading Material

Text Books

1. Brian W. Kernighan, Dennis M. Ritchie. "The C Programming Language, 2nd Edition," Prentice-Hall India.

Additional Readings

1. G. Michael Schneider. "Introductionto Programmingand Problem SolvingwithPASCAL," JohnWiley and Sons.
2. Brian W. Kernighan and R. Pike. "The Unix Programming Environment," Prentice-Hall India.
3. Chakravarthy Bhagvati. "How to Program (An Informal Guide)," http://dcis.uohyd.ernet.in/ ~chakcs/howtoprogram.pdf

Suggested Assignments

1. Write a program for *Collatz Conjecture*.
   Input: An integer x($< 10^7$) **Problem:** If x is *even*, x = x/2 If x is *odd*, then x =3x +1 Repeat the above step untilx =1. **Output:** The number of times the above step is repeated until x =1.
2. Write a program to verify if an input string is a valid vehicle number.
   Note that a vehicle number is of the form: CCDDCDDDD or CCDDCCDDDD where 'C' stands for a single letter and 'D' stands for a digit.
3. Read "AdditionalReading3" carefully; follow the steps given there and writea program to produce a 4 × 4 Magic Square.