Lab 01:
Problem solving and Algorithms

Objective(s): Learn about problem solving and algorithms.

CLOs: CL01, CLO4

In computing, we focus on the type of problems categorically known as algorithmic problems, where their solutions are expressible in the form of algorithms.

An algorithm a well-defined computational procedure consisting of a set of instructions that takes some value or set of values, as input, and produces some value or set of values, as output. In other word, an algorithm is a procedure that accepts data; manipulate them following the prescribed steps, so as to eventually fill the required unknown with the desired value(s).

Tersely put, an algorithm, a jargon of computer specialists, is simply a procedure. People of different professions have their own form of procedure in their line of work, and they call it different names. A cook, for instance, follows a procedure commonly known as a recipe that converts the ingredients (input) into some culinary dish (output), after a certain number of steps.

An algorithm is a form that embeds the complete logic of the solution. Its formal written version is called a program, or code. Thus, algorithmic problem solving actually comes in two phases: derivation of an algorithm that solves the problem, and conversion of the algorithm into code. The latter, usually known as coding, is comparatively easier, since the logic is already present – it is just a matter of ensuring that the syntax rules of the programming language are adhered to. The first phase is what that stumbles most people, for two main reasons. Firstly, it challenges the mental faculties to search for the right solution, and secondly, it requires the ability to articulate the solution concisely into step- by-step instructions, a skill that is acquired only through lots of practice. Many people are able to make claims like “oh yes, I know how to solve it”, but fall short when it comes to transferring the solution in their head onto paper.

Algorithms and their alter ego, programs, are the software. The machine that runs the programs is the hardware.

Pseudocode is a method of describing computer algorithms using a combination of natural language and programming language. It is essentially an intermittent step towards the development of the actual code. It allows the programmer to formulate their thoughts on the organization and sequence of a computer algorithm without the need for actually following the exact coding syntax. Although pseudocode is frequently used there are no set of rules for its exact implementation. In general, here are some rules that are frequently followed when writing pseudocode:

Symbols are used for arithmetic operations (+, -, *, /, **).
Symbolic names are used to indicate the quantities being processed.
Certain keywords can be used, such as PRINT, WRITE, READ, INPUT, OUTPUT etc.
Indentation should be used to indicate branches and loops of instruction.

Examples of Algorithms:

Write an algorithm to find area of a rectangle

Step 1: Start
Step 2: get l and b values
Step 3: Calculate A = l * b
Step 4: Display A
Step 5: Stop

Write an algorithm for Calculating area and circumference of circle

Step 1: Start
Step 2: get r value
Step 3: Calculate A = 3.14 *r * r
Step 4: Calculate C = 2.3.14 * r
Step 5: Display A, C
Step 6: Stop

To check greatest of two numbers

Step 1: Start
Step 2: get num1 and num2 value
Step 3: check if (a>b) print num1 is greater
Step 4: else num2 is greater
Step 5: Stop

Lab Tasks:

1. Write an algorithm to add two numbers
2. Write an algorithm to find velocity of a moving object
3. Write an algorithm to check positive or negative number
4. Write an algorithm to find volume of cylinder