

Algorithm Development and Programming Fundamentals

What is Programming?

What is Programming?

It is a set of instructions given in specific manner to perform a specific task.

It can be of any type...

- Making Tea
- Cooking Rice
- Driving Car

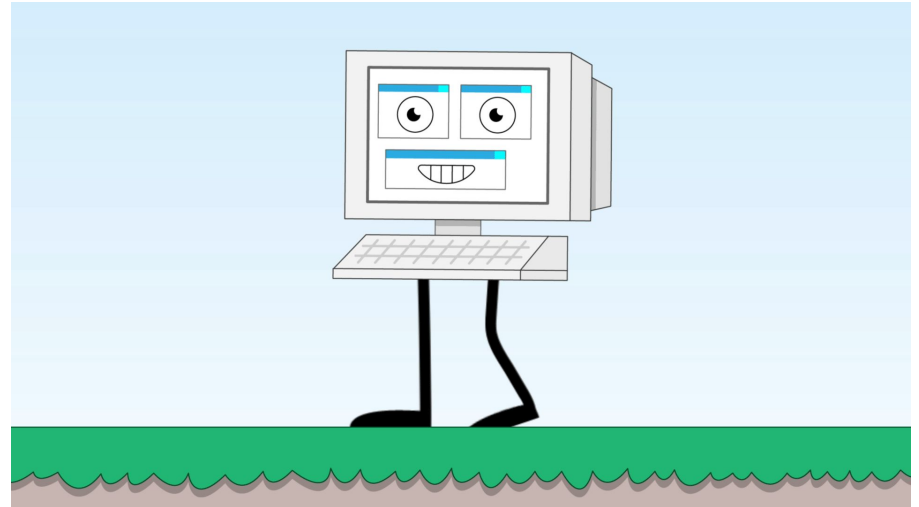
Instructions can also be given to computers.



Computer Programming

Computer programming is the act of writing computer programs, which are a sequence of instructions written using a Computer Programming Language to perform a specified task by the computer.

- a. left foot forward
- b. right foot forward
- c. go back to instruction a



What is Programming?

Programming is the process of taking an algorithm and encoding it into a notation, a programming language, so that it can be executed by a computer.

Although many programming languages and many different types of computers exist, the important first step is the need to have the solution.

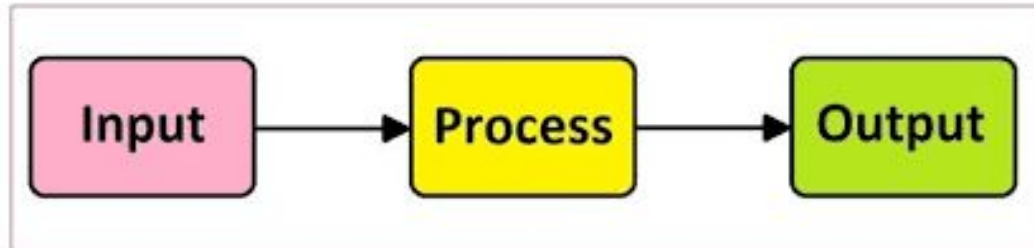
Without an algorithm there can be no program.

Processing of Data

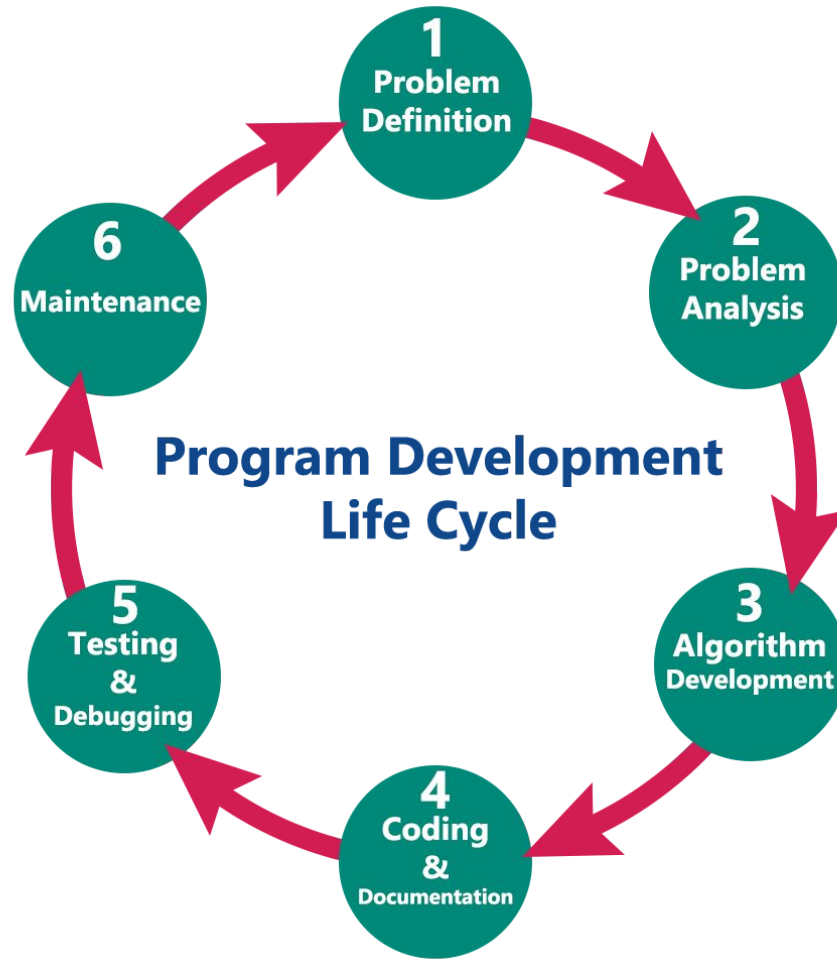
What is Information Processing

The input–process–output (IPO) model is a widely used approach in systems analysis and software engineering for describing the structure of an information processing program or another process.

1. A requirement from the environment (input)
2. A computation based on the requirement (process)
3. A provision for the environment (output)



Program Development Life Cycle



PDLC

1. Analyze the problem.
2. Design the program.
3. Code the program.
4. Debug the program.

Defining the Problem

Problem Definition

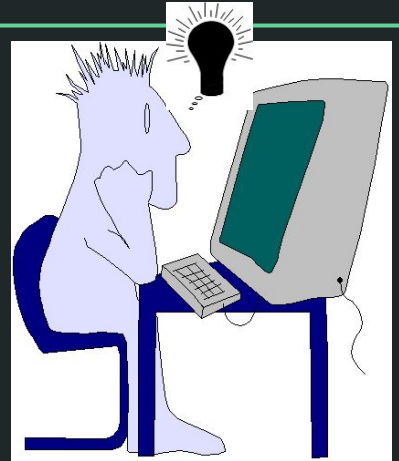
In this phase, we define the problem statement and we decide the boundaries of the problem.

In this phase we need to understand the problem statement, what is our requirement, what should be the output of the problem solution.

These are defined in this first phase of the program development life cycle.

The program specification defines the data used in program, the processing that should take place while finding a solution, the format of the output and the user interface.

Analyze the Problem



Problem Analysis

In this phase we determine the requirements like variables, functions, etc. to solve the problem.

That means we gather the required resources to solve the problem defined in the problem definition phase.

We also determine the bounds of the solution.

Algorithm Development

Develop Algorithm

During this phase, we develop a step by step procedure to solve the problem using the specification given in the previous phase.

This phase is very important for program development.

That means we write the solution in step by step statements.

Program design starts by focusing on the main goal that the program is trying to achieve and then breaking the program into manageable components, each of which contributes to this goal.

This approach of program design is called top-bottom program design or modular programming.

The first step involve identifying main routine, which is the one of program's major activity. From that point, programmers try to divide the various components of the main routine into smaller parts called modules.

For each module, programmer draws a conceptual plan using an appropriate program design tool to visualize how the module will do its assign job.

Tools

Structure Charts – A structure chart, also called Hierarchy chart, show top-down design of program. Each box in the structure chart indicates a task that program must accomplish. The Top module, called the Main module or Control module.

Algorithms – An algorithm is a step-by-step description of how to arrive at a solution in the most easiest way. Algorithms are not restricted to computer world only. In fact, we use them in everyday life.

Flowcharts – A flowchart is a diagram that shows the logic of the program.

Tools

Decision tables – A Decision table is a special kind of table, which is divided into four parts by a pair of horizontal and vertical lines.

Pseudocode – A pseudocode is another tool to describe the way to arrive at a solution. They are different from algorithm by the fact that they are expressed in program language like constructs.

Coding and Documentation

Coding the Solution

This phase uses a programming language to write or implement the actual programming instructions for the steps defined in the previous phase.

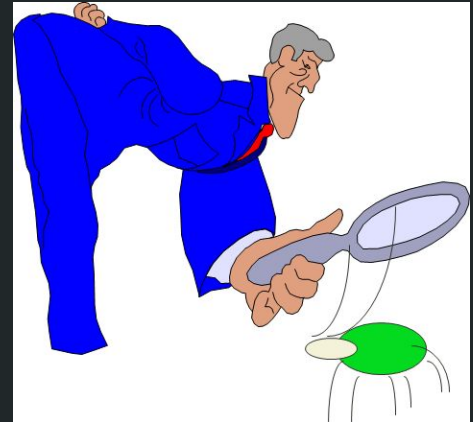
In this phase, we construct the actual program.

That means we write the program to solve the given problem using programming languages like C, C++, Java, etc.,

Coding the program means translating an algorithm into specific programming language.

The technique of programming using only well defined control structures is known as Structured programming.

Test the Solution



Testing

During this phase, we check whether the code written in the previous step is solving the specified problem or not.

That means we test the program whether it is solving the problem for various input data values or not.

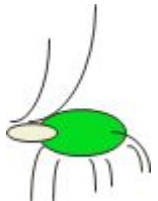
We also test whether it is providing the desired output or not.

After removal of syntax errors, the program will execute.

However, the output of the program may not be correct. This is because of logical error in the program.

A logical error is a mistake that the programmer made while designing the solution to a problem. So the programmer must find and correct logical errors by carefully examining the program output using Test data.

Syntax error and Logical error are collectively known as Bugs. The process of identifying errors and eliminating them is known as Debugging.



Maintenance

Maintenance

During this phase, the program is actively used by the users.

If any enhancements found in this phase, all the phases are to be repeated to make the enhancements.

That means in this phase, the solution (program) is used by the end-user. If the user encounters any problem or wants any enhancement, then we need to repeat all the phases from the starting, so that the encountered problem is solved or enhancement is added.

Algorithms

Introduction

The set of sequential steps developed to solve a problem is called its Algorithm.

And the implementation of algorithm in high computer programming language is called a Program.

Generally every algorithm is converted into a graphical format called Flow Chart before it is implemented as program.

The word algorithm comes from the word Al-khowarizmi, an Arabian inventor. It means recipe, method technique or procedure.

Properties

An algorithm is a step-by-step problem solving procedure that can be carried out by a computer.

- It should be simple
- It should be clear with no ambiguity
- It should lead to unique solution of the problem
- It should involve a finite number of steps to arrive at a solution
- It should have the capability to handle some unexpected situations which may arise during the solution of a problem

Characteristics/ Features

1. Input
 2. Process Method
 3. Finiteness
 4. Effective basic instructions
 5. Output
-
- Adaptability on a computer
 - Simplicity of logic
 - Elegance
 - Efficiency with which it can be exacted

Steps to develop algorithm

1. Clearly understand the problem statement.
2. Study the outputs to be generated
3. Design the process
4. Refine the process
5. Test the algorithm

Advantages

1. Easy to understand as it is a step by step process
2. It has got a definite procedure
3. Convenient to develop an algorithm and then convert it a flow chart and eventually an program.
4. It has a start and end which simplifies the procedure
5. Easy to debug
6. Independent of programming languages




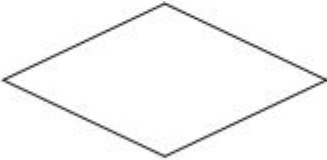


Disadvantage

It is time consuming and cumbersome.

Flow Charts

- A flowchart is a blueprint that pictorially represents the algorithm and its steps.
- A flowchart is a diagram that depicts a process, system or computer algorithm.
- Flowcharts use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence.
- They can range from simple, hand-drawn charts to comprehensive computer-drawn diagrams depicting multiple steps and routes.
- Benefits of Flowchart are: Simplify the Logic, Better Communication, Effective Analysis, Useful in Coding, Testing, Documentation

Symbols

Name	Symbol	Use in Flowchart
Oval		Denotes the beginning or end of the program
Parallelogram		Denotes an input operation
Rectangle		Denotes a process to be carried out e.g. addition, subtraction, division etc.
Diamond		Denotes a decision (or branch) to be made. The program should continue along one of two routes. (e.g. IF/THEN/ELSE)
Hybrid		Denotes an output operation
Flow line		Denotes the direction of logic flow in the program



Process symbol

Also known as an “Action Symbol,” this shape represents a process, action, or function. It’s the most widely-used symbol in flowcharting.



Start/End symbol

Also known as the “Terminator Symbol,” this symbol represents the start points, end points, and potential outcomes of a path. Often contains “Start” or “End” within the shape.



Document symbol

Represents the input or output of a document, specifically. Examples of an input are receiving a report, email, or order. Examples of an output using a document symbol include generating a presentation, memo, or letter.



Decision symbol

Indicates a question to be answered — usually yes/no or true/false. The flowchart path may then split off into different branches depending on the answer or consequences thereafter.



Connector symbol

Usually used within more complex charts, this symbol connects separate elements across one page.



Off-Page
Connector/Link
symbol

Frequently used within complex charts, this symbol connects separate elements across multiple pages with the page number usually placed on or within the shape for easy reference.



Input/Output
symbol

Also referred to as the “Data Symbol,” this shape represents data that is available for input or output as well as representing resources used or generated. While the paper tape symbol also represents input/output, it is outdated and no longer in common use for flowchart diagramming.



Comment/Note
symbol

Placed along with context, this symbol adds needed explanation or comments within the specified range. It may be connected by a dashed line to the relevant section of the flowchart as well.

Pseudo code

- Pseudo-code is an informal way to express the design of a computer program or an algorithm.
- The aim is to get the idea quickly and also easy to read without syntactic details.
- Pseudocode is not a real language of programming.
- It can not thus be compiled into a software executable.
- It is used to construct code for programmes in brief phrases or basic English syntaxes prior to it becoming a certain programming language.
- This is done to identify the flow errors in the highest possible level and comprehend the data flows that will be used by the final software.

- This helps undoubtedly to save time when conceptual problems are already fixed during real programming.
- To get the desired results for a program, you first gather the program description and functionality, then use pseudo code for creating statements.
- Detailed pseudocode is evaluated and validated according to design standards by the design team or programmers.
- Catching faults or incorrect program flows during the phase of pseudo code creation is useful since it is cheaper than catching them afterwards.
- Once the team accepts the pseudo code, it is written in a programming language using vocabulary and grammar.

Examples

Find area of a rectangle - Algorithm

Step 1: Start

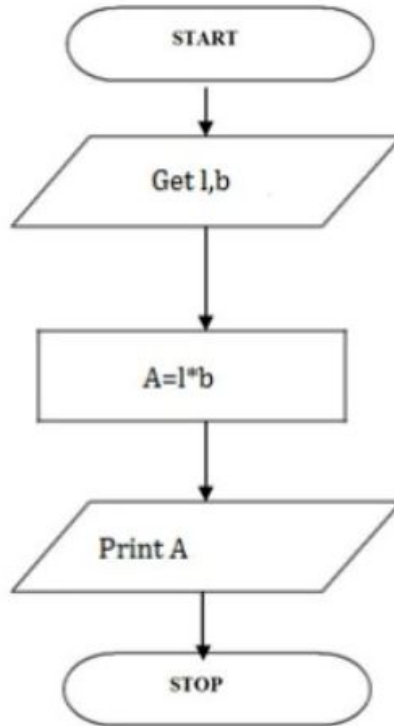
Step 2: get l,b values

Step 3: Calculate $A=l*b$

Step 4: Display A

Step 5: Stop

Find area of a rectangle - Flowchart



Find area of a rectangle - Psuedocode

BEGIN

READ l, b

CALCULATE $A = l * b$

DISPLAY A

END

Greatest of two numbers - Algorithm

Step 1: Start

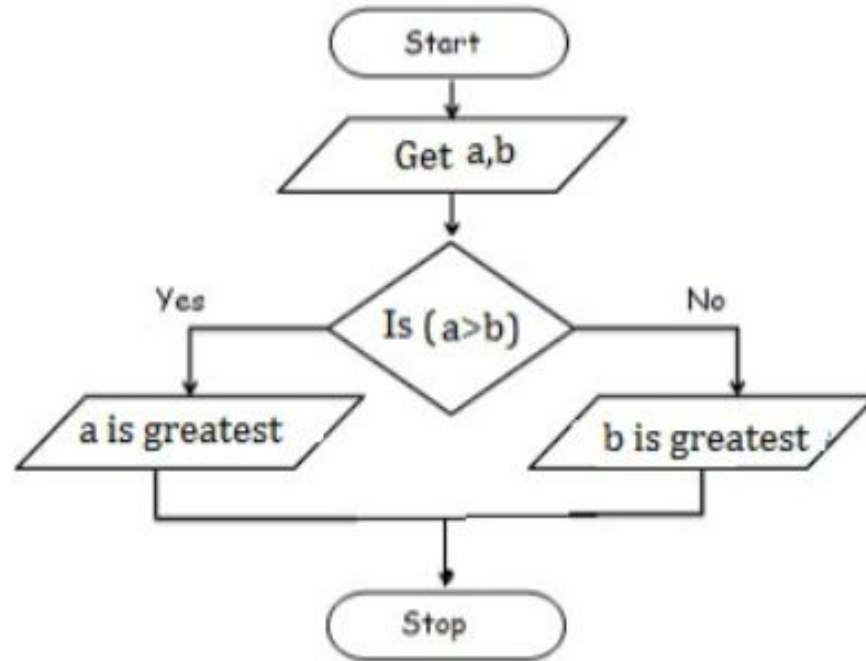
Step 2: get a,b value

Step 3: check if($a > b$) print a is greater

Step 4: else b is greater

Step 5: Stop

Greatest of two numbers - Flowchart



Greatest of two numbers - Psuedocode

```
BEGIN  
  READ a,b  
  IF (a>b) THEN  
    DISPLAY a is greater  
  ELSE  
    DISPLAY b is greater  
  END IF  
END
```