

Unit 1

Introduction to computer System and Programming

Computer:

Programmable machine designed to perform arithmetic and logical operation automatically and sequentially as inputs given by users.

What are software, hardware and Operating Systems?

There are very clear differences between software, hardware and an operating system that are important to understand when learning about computer science.

Software is **program that can be run on a computer or set of instructions run by a computer'**

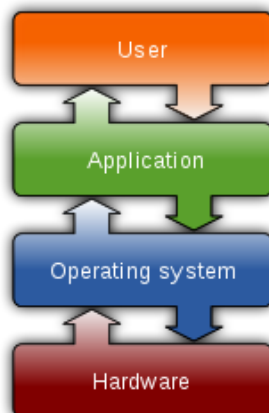
Hardware is the **physical parts of a computer, such as the processor, memory modules and the screen.**

The **Operating System** is often described as a translator; it **translates the language of the hardware (binary numbers) into the language of the software (written programs) and then displays it in a way that humans can understand (text, images and sound).**

Refer the video : <https://youtu.be/AkFi90lZmXA>

Operating System

An **operating system (OS)** is system software that manages computer hardware and software resources and provides common services for computer programs.



Input/output Devices

Basic hardware components of a computer are introduced:

- **Inputs are any devices that send information into the computer.** For example, a mouse has a position on the screen and is able to tell the computer when a button is clicked and which button is clicked. A keyboard is able to send key presses to the computer.
- **Outputs are any devices that are able to show information to the user.** Good examples of these are the screen, which shows the user text and images, and headphones, which are able to play sounds for the user to listen to.

CPU / Processor

- **A processing device is any device that is used to run calculations using binary numbers. The main processing device in any computer is the CPU (Central Processing Unit),** which performs binary calculations to make the computer run. This component is often referred to as the ‘brain of the computer’.

Storage Devices

- **Storage devices** hold information and are often referred to as computer memory. There are two main types of memory: persistent memory (that exists when the computer is switched off) and volatile memory (that is deleted when the computer turns off). An example of a storage device would be a hard drive or a USB memory stick.

Inputs	Outputs	Processing	Storage
Keyboard	Screen	CPU	USB memory
Mouse	Headphones		Hard Drive
Webcam	Printer		RAM

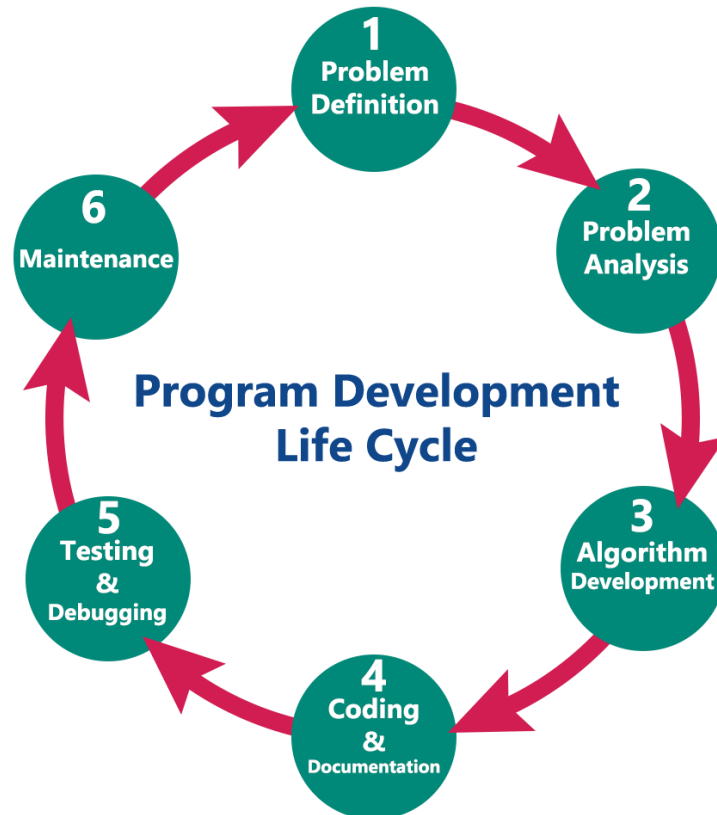
Program Development Cycle

When we want to develop a program using any programming language, we follow a sequence of steps. These steps are called phases in program development. The program development life cycle is a set of steps or phases that are used to develop a program in any programming language.

Generally, program development life cycle contains 6 phases, they are as follows....

- Problem Definition
- Problem Analysis
- Algorithm Development

- Coding & Documentation
- Testing & Debugging
- Maintenance



1. 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.

2. Problem Analysis

In phase 2, 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.

3. Algorithm Development

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.

4. Coding & Documentation

This phase uses a programming language to write or implement actual programming instructions for the steps defined in the previous phase. In this phase, we construct actual program. That means we write the program to solve the given problem using programming languages like C, C++, Java etc.,

5. Testing & Debugging

During this phase, we check whether the code written in 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 that whether it is providing the desired output or not.

6. 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 again 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.

Algorithm

An algorithm is a sequence of well-defined instructions for completing a task or solving a problem. It can be described in a natural language, pseudocode, a flowchart, or even a programming language. For example, suppose we are interested in knowing whether a specific number is contained in a given sequence of numbers.

al·go·rithm [al-guh-rith-uhm] Noun

A set of rules for solving a problem in a finite number of steps.

Algorithm Examples:

You might need algorithms to:

1. Convert from one unit to another. (feet, yards, kilometers, miles)
2. Find square of Number.
3. Print total income from items sold in the week.

Example of Algorithm:

Algorithm 1 : Algorithm to convert temperature from Fahrenheit to Celsius.

Input: F

Output : C

F : temperature in Fahrenheit

C : temperature in Celsius.

Step 1: Start.

Step 2: Read temperature (Fahrenheit) in F.

Step 3: calculate $C=(5(F-32))/9$.

Step 4: Display C as temperature in Celsius.

Step 5: Stop.

Algorithm Defined

A step-by-step procedure to perform a task.

Examples:

- ✓ Sort a list of names in alphabetical order
- ✓ Quickly find a name in a sorted list
- ✓ Merge two alphabetical lists into one alphabetical list
- ✓ Find the greatest common divisor (GCD) of two integers.

Summary

- Algorithm: A set of rules to solve a problem.
- Program Development Steps help us to organize a solution.



Characteristics or properties of Algorithm:

Finiteness: The algorithm must always terminate after a finite number of steps.

Definiteness: Each step must be precisely defined; the actions to be carried out must be unambiguously specified for each case.

Input: An algorithm has zero or more inputs, taken from a specified set of objects.

Output: An algorithm has one or more outputs, which have a specified relation to the inputs.

Effectiveness: All operations to be performed must be sufficiently basic that they can be done exactly and in finite length.

OR

1. **Precision** – the steps are precisely stated (defined).
2. **Uniqueness** – results of each step are uniquely defined and only depend on the input and the result of the preceding steps.
3. **Finiteness** – the algorithm stops after a finite number of instructions are executed.
4. **Input** – the algorithm receives input.
5. **Output** – the algorithm produces output.
6. **Generality** – the algorithm applies to a set of inputs.

Advantages of Algorithms:

1. It is a step-wise representation of a solution to a given problem, which makes it easy to understand.
2. An algorithm uses a definite procedure.
3. It is not dependent on any programming language, so it is easy to understand for anyone even without programming knowledge.
4. Every step in an algorithm has its own logical sequence so it is easy to debug.
5. By using algorithm, the problem is broken down into smaller pieces or steps hence, it is easier for programmer to convert it into an actual program.

Disadvantages of Algorithms:

1. Algorithm is Time consuming.
2. Difficult to show Branching and Looping in Algorithms.
3. Big tasks are difficult to put in Algorithms.

References

<https://www.owlgen.com/question/what-is-an-algorithm-explain-characteristics-of-an-algorithm-with-the-help-of-an-example>

http://btechsmartclass.com/c_programming/C-Program-Development-Life-Cycle.html

http://courses.cs.vt.edu/cs2104/Fall12/notes/T16_Algorithms.pdf