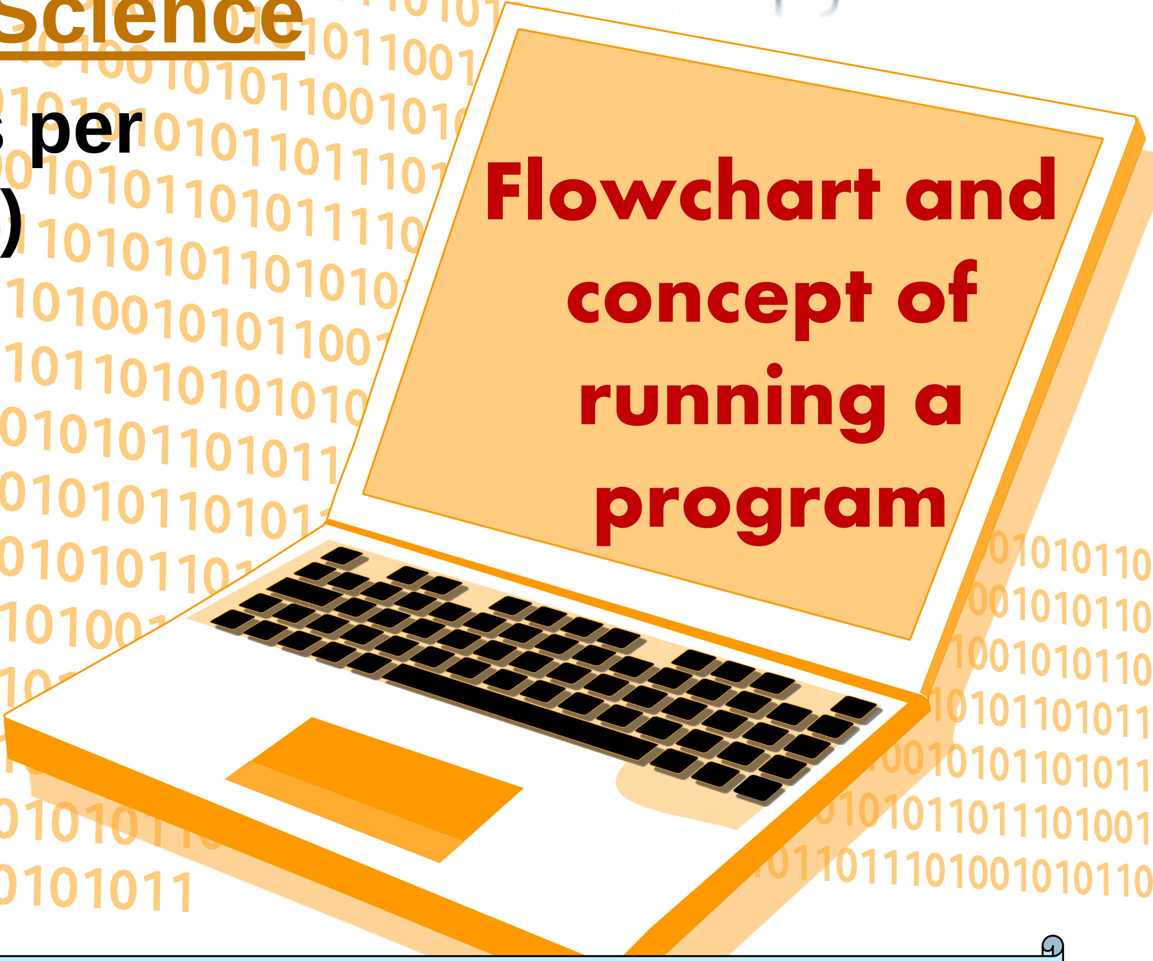
Chapter 3 :

Computer Science

Class XI (As per CBSE Board)



Flowchart and concept of running a program

New Syllabus 2019-20

Visit : python.mykvs.in for regular updates

Flowchart

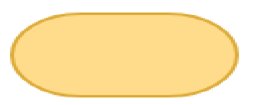
A flowchart is simply a graphical representation of steps. It shows steps in a sequential order, and is widely used in presenting flow of algorithms, workflow or processes. Typically, flowchart shows the steps as boxes of various kinds, and their order by connecting them with arrows.

Flowchart Symbols

Different flowchart shapes have different conventional meanings. The meanings of some of the more common shapes are as follows:

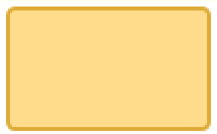
1. Terminator

The terminator symbol represents the starting or ending point of the system.



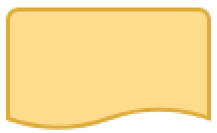
2. Process

A box indicates some particular operation.



3. Document

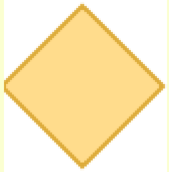
This represents a printout, such as a document or a report.



Flowchart

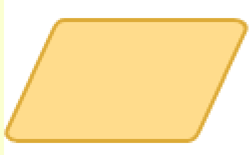
4. Decision

A diamond represents a decision or branching point. Lines coming out from the diamond indicates different possible situations, leading to different sub-processes.



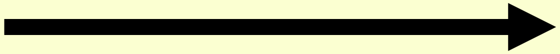
5. Data

It represents information entering or leaving the system. An input might be an order from a customer. An output can be a product to be delivered.



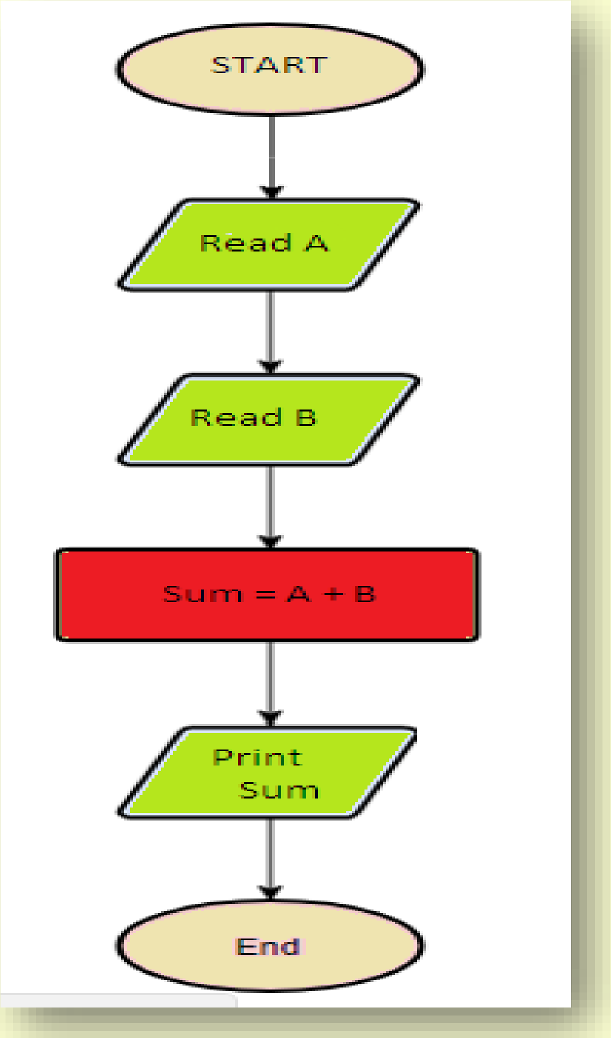
6. Flow

Lines represent flow of the sequence and direction of a process.



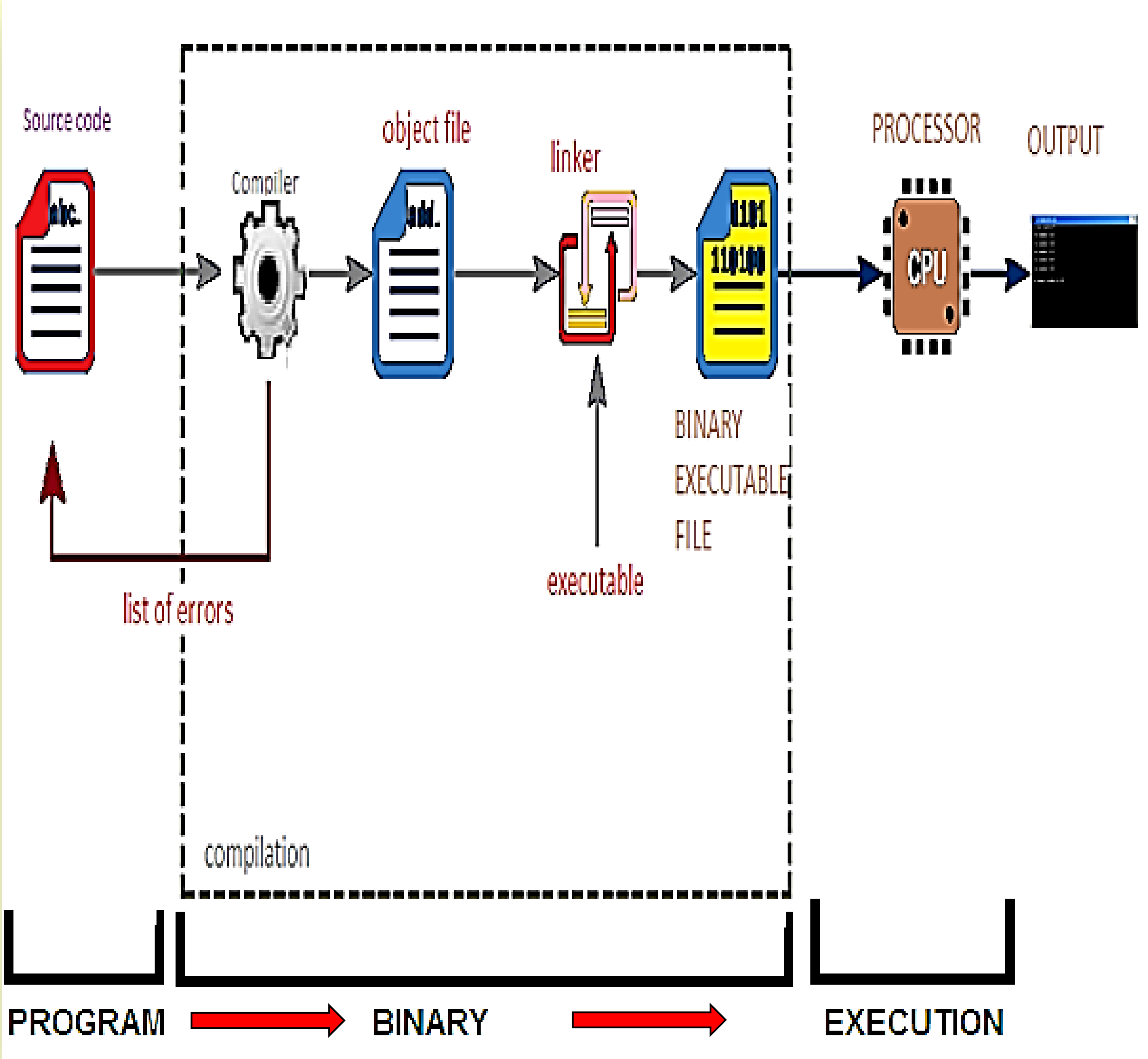
Flowchart

Flowchart for addition of two numbers



Running a program

Generally, the programs developed in high level language like C, C++, Java etc.,cannot understand by the computer . It can understand only low level language. So, the program written in high level language to be converted into low level language to make it understandable for the computer. This conversion is performed using either Interpreter or Compiler.The basic flow of any program execution is shown in diagram.



Running a Program

The compilation mainly undergoes following steps

1. **Processing phase->** removal of comments and carry out of processor directives
2. **Compilation phase ->**

a.) analysis phase-identification of tokens

b.)synthesis phase – formation of syntax tree

3. Assembly phase -> source code is converted into object code(a form of machine code)

4. Linking -> linking of other executable files with object code

5. Loader -> This program Loads the executable module into memory for execution.

Difference between Compiler an interpreter

Interpreter	Compiler
Translates program one statement at a time.	Scans the entire program and translates it as a whole into machine code.
It takes less amount of time to analyze the source code but the overall execution time is slower.	It takes large amount of time to analyze the source code but the overall execution time is comparatively faster.
No intermediate object code is generated, hence are memory efficient.	Generates intermediate object code which further requires linking, hence requires more memory.
Continues translating the program until the first error is met, in which case it stops. Hence debugging is easy.	It generates the error message only after scanning the whole program. Hence debugging is comparatively hard.
Programming language like Python, Ruby use interpreters.	Programming language like C, C++ use compilers.

Operating System as a Resource Manager

There are no. of resources available in a computer system

- **CPU**
- **Memory**
- **i/o devices**

If a computer system is used by multiple applications then they will compete for these resources
It is the job of the Operating System to allocate these resources to the various applications so that :

- **The resources are allocated fairly**
- **The resources are protected from cross access**
- **Access to the resources is synchronized so that operations are correct and consistent**
- **Deadlocks are detected , resolved and avoided.**

Major Operating System Functions

1. Process Management

CPU can perform one task at one time. if there are many tasks, operating system decides which task should get the CPU.

2. Memory Management

if there is no operating system, the programs may mix with each other. The OS manages Memory to not to mix programs and to provide memory to these programs.

3. Input/Output management -operating system controls all the inputs and outputs devices, monitors their request and issues commands to the devices.

Program execution under Operating System(Process Management) / How an operating system runs a program

Any Program under execution of processor is known as a process. A process includes the complete execution context (code to execute, data to manipulate, registers, OS resources in use). Following are the major activities of an operating system with respect to program management −

- ❑ Loads a program into memory.
- ❑ Executes the program.
- ❑ Handles program's execution.
- ❑ Provides a mechanism for process synchronization.
- ❑ Provides a mechanism for process communication.
- ❑ Provides a mechanism for deadlock handling.

Process states diagram

A process changes its state during its execution. Each process may be in one of the following states:

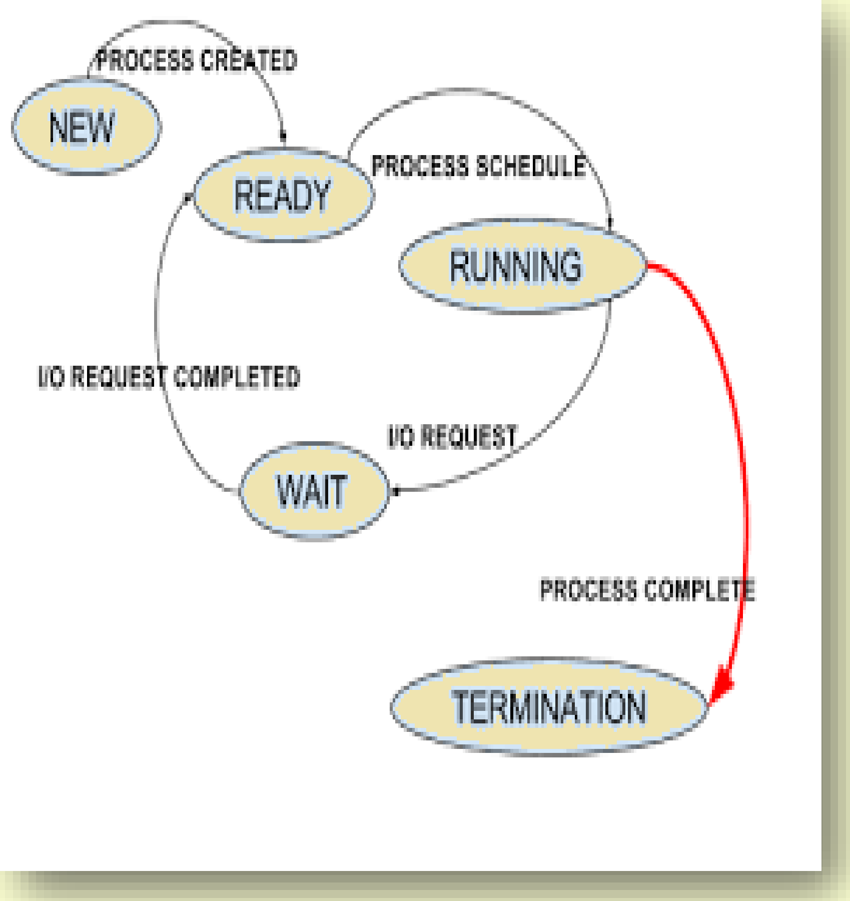
New: when a new process is being created.

Running: A process is said to be in running state when instructions are being executed.

Waiting: The process is waiting for some event to occur (such as an I/O operation).

Ready: The process is waiting for processor.

Terminated: The process has finished execution.



Process Control Block

It is a data structure maintained by the Operating System for every process. The PCB is identified by an integer process ID (PID). A PCB keeps all the information needed to keep track of a process.

Process ID-Unique identification for each of the process in the operating system.

Process State-The current state of the process i.e., whether it is ready, running, waiting, or whatever.

Process privileges-This is required to allow/disallow access to system resources.

Pointer-A pointer to parent process.

Program Counter-Program Counter is a pointer to the address of the next instruction to be executed for this process.

CPU registers-Various CPU registers where process need to be stored for execution for running state.

CPU Scheduling Information-Process priority Information

Process ID
Process State
Pointer
Process priority
Program Counter
CPU registers
Etc.

A typical Process Control Block

Process Scheduling

Nowadays Processor Performs multiple task at a time.It is only possible when each task(process) is scheduled.There are two types of scheduling.

1) **Preemptive**: In this all the Processes are executed by using some Amount of Time of CPU.

2) **NON-Primitive**: In **this No Time Scheduling** is used. after Completing the First Process, this will Automatically start the Second Process.

Some scheduling techniques are following-

1) **First Come First Serve**: As the name Suggest, the Processes those are Coming first, will be Executed first

2) **Shortest Job first:** In this Scheduling, All the Process are Arranged into their Size and shortest sized process is selected first.

3) **Priority Scheduling**: When the Process are Given, then Each Process have a Priority.Highest priority is choosen first.

4) **Round Robin**: In this Scheduling the Time of CPU is divided into the Equal Parts and Assign to various Processes.

5) **Multilevel Queue Scheduling**: In this The Time of CPU is divided by using Some Process Categories.