



COMSATS University
Islamabad
(Lahore Campus)

CSC103- Programming Fundamentals

MS. MAHWISH WAQAS

MAWISH.WAQAS@CUILAHORE.EDU.PK

Lecture 1: Introduction to Programming

Outline

- **What are computers?**
 - **What is software?**
 - **What is programming?**
 - **Understanding the steps to write/run programs**
 - **Basic Concepts**
 - **Algorithms**
 - Pseudo code
 - Flowcharts
 - **Low and high level languages**
 - **First C program**
-

What is a Computer?

Computer

- Device capable of **performing computations** and making **logical decisions**
- Computers process data under the control of sets of instructions called **computer programs**

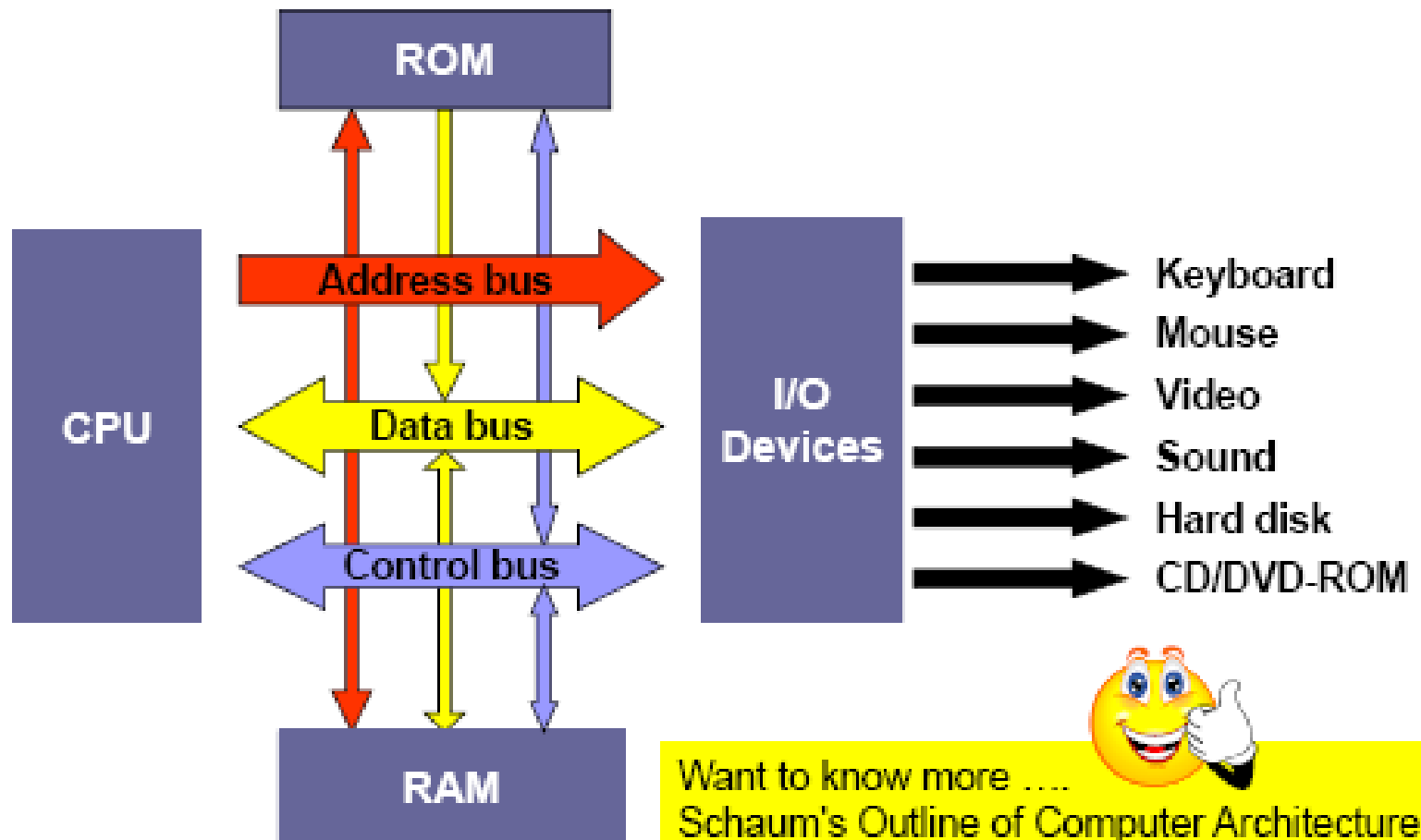
Hardware

- Various devices comprising a computer
- Keyboard, screen, mouse, disks, memory, CD-ROM, and processing units

Software

- Programs that run on a computer
 - System Software
 - Application Software

What's Inside a Computer



Computer!

Tablet

Palmtop

Laptop

Desktop

And many more!!!

System Software vs Application Software

System software (OS)

- Serving all users
- I.e., everyone needs it
- Known functionalities

Application software (MS Office)

- Serving those users who need the specific functions
- I.e., **not** everyone needs it
- User-specific functionalities

Examples of application software

MS Office

Internet Explorer

Real Player

Whatever application you can think of!

Too Many of These

Can't talk about them all

- Select some

But which?

- Some want to learn this; some want that

How applications are created

Programmer's Secret Life

- Terminology
- Program composition
- Programming languages
- Programming language classification

Terminology

Application

- Short for application software
- Short for application program

Programme (noun)

- Specific, countable

Software

- General, not countable
- Usually meant a **collection** of programs

3P's

Programme

- A set of rules for your computers to follow in order to achieve a goal

Programmer

- A person who produces the program
- A person who makes a living from producing programs

Programming language

- A way for a programmer to write about the set of rules

As If

Computers are creatures from outer space

They speak strange languages

Programmers study their languages

Therefore, they are able to tell the computers what to do

The 3P Relationship

Programmers write programs
using programming languages.

In Literature

Ashfaq Ahmed writes Funkar using Urdu.

In Computer Science

Ali writes Hello World
using C.

It is a process of
creativity.

Writing vs. Programming

- ▶ Setting the theme
- ▶ Structuring
- ▶ Writing
- ▶ Proof-reading
- ▶ Defining the problem
- ▶ Planning the solution
- ▶ Coding the program
- ▶ Testing the program
- ▶ **Documenting the program**
 - commenting

Defining The Problem

Input

Output

Problem to be solved

Planning The Solution Algorithms

Computing problems

- All can be solved by executing a series of actions in a specific order

Algorithm: procedure in terms of

- Actions to be executed
- The order in which these actions are to be executed

Program control

- Specify order in which statements are to be executed

Algorithm

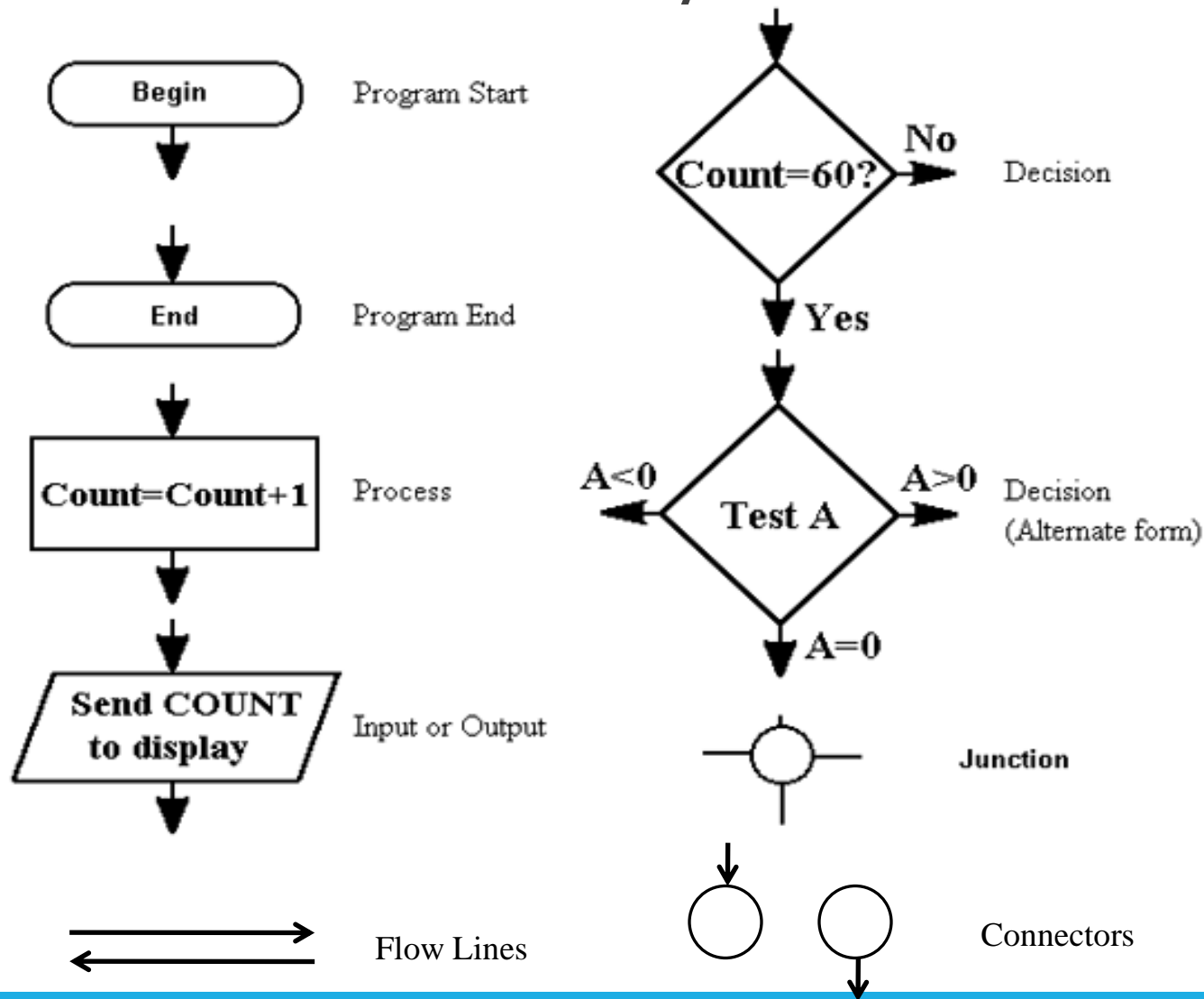
- The **order of the actions** in an algorithm is important
 - Example: algorithm to go to work every morning
 - Get out of bed;
 - Take a bath;
 - Get dressed;
 - Eat breakfast;
 - Take the bus to work.
- If I carry out the same actions in a different order
 - Get out of bed;
 - Get dressed;
 - Take a bath;
 - Eat breakfast;
 - Take the bus to work.
- Then, I will still get to work, however soaking wet

So order is important

Pseudo Code and Flowchart

- However, programming languages can be **complex** for students that are starting to learn how to program
- The focus in programming fundamentals is also to learn **how to solve a problem** through a computer program
- Therefore, we will introduce the use of **pseudo codes** and **flowcharts** to describe algorithms (solutions of the proposed problems) and after that, we will learn how to translate them to a programming language, like C

Flow chart Symbols

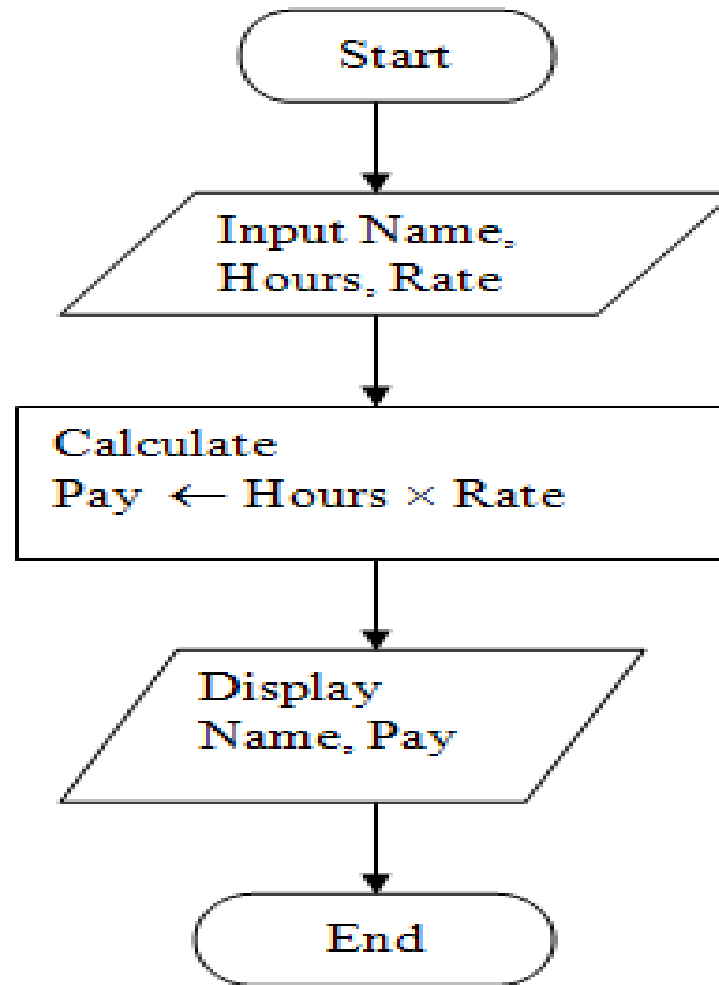


Pseudo Code

Example: The following set of instructions forms a detailed algorithm in pseudo-code for calculating the payment of person.

- Begin
- Input the three values into the variables Name, Hours, Rate.
- Calculate $\text{Pay} = \text{Hours} * \text{Rate}$.
- Display Name and Pay.
- End

Flowcharts



Flow Chart another Example

(b) Pseudo code

Place 0 in sum

Place 0 in counter

Enter first number

DOWHILE the number is not equal to 999

 Add number to sum

 Add 1 to counter

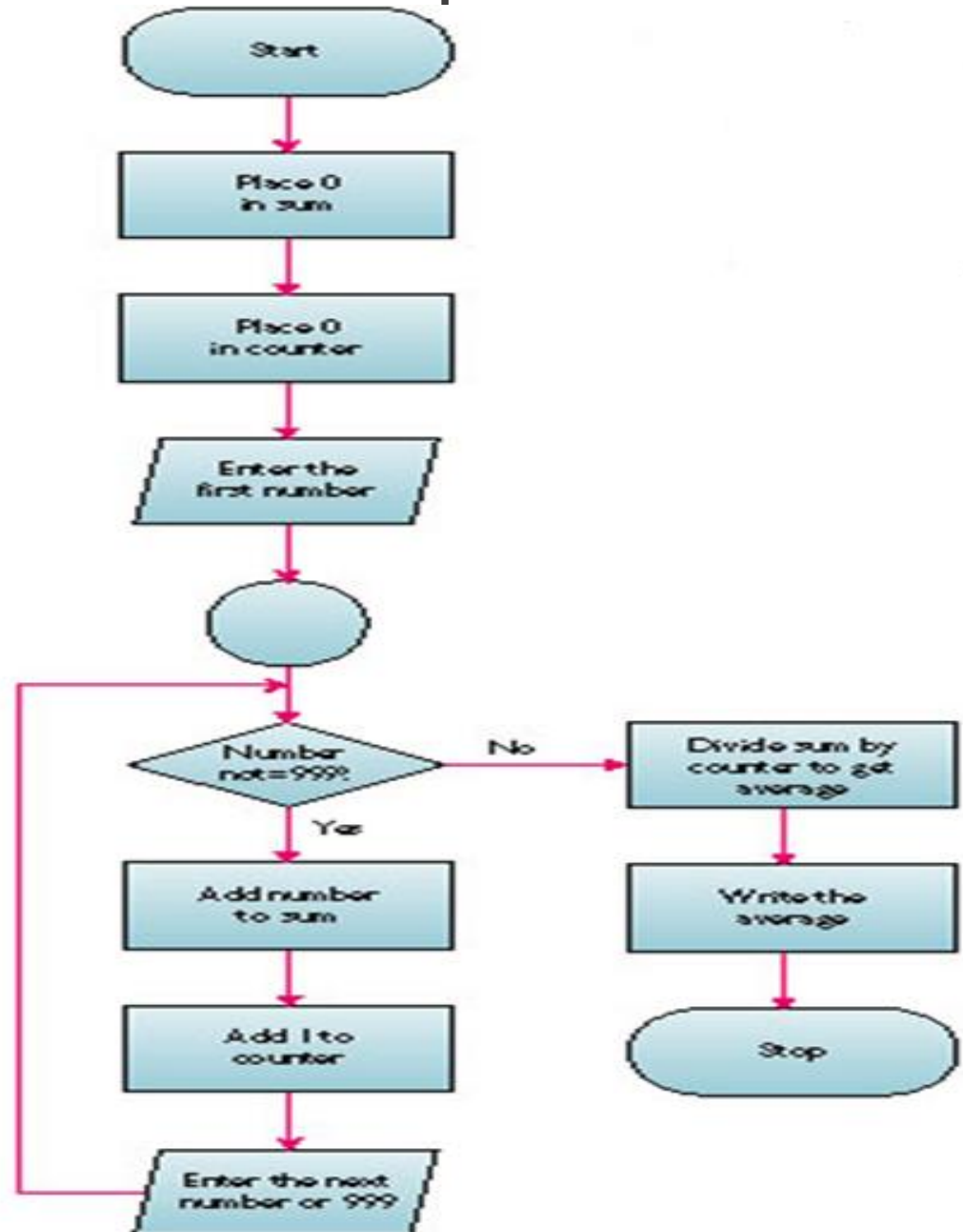
 Enter next number or 999

ENDDO

Divide sum by counter to get average

Write the average

Accept series of numbers
and display the average



What is Computer Programming?

A ***way to solve problems*** using a computer

How?

- Through a sequence of very clear and defined steps that aim to solve the problem -> this is called an **algorithm**

Algorithms are not only used for computer programming, they are a ***general concept*** for problem solving

Coding The Program

Sentence by sentence, word by word

Detailed description in the chosen language

Testing The Program

- ▶ Translation – compiler
 - Translates from **source** module into **object module**
 - Detects syntax errors
- ▶ Link – linkage editor (linker)
 - Combines **object module** with libraries to create **load module**
 - Finds undefined external references (run-time errors)
- ▶ Debugging
 - Run using data that tests all statements
 - Logic errors

Distinction

Source code

- Your creation

Object modules

- Machine code of your creation

Load modules

- Machine code of pre-installed functions

Executable programs

- Combination of your and pre-installed machine code

Documenting The Program

Comments within source code

In plain English

Convenient for

- The programmer him/herself who needs to read the program later
- Somebody else who needs to read the program

Executing a Program

Does the computer understand the C language?

- 'C' is a high level computer language
- A computer only understands patterns of 0s and 1s

'C' allows you to write a computer program using statements that are much easier for you to understand than 0s and 1s

Compilers vs. Interpreters

Compilers

- Compile several machine instructions into short sequences to simulate the activity requested by a single high-level primitive
- Produce a machine-language copy of a program that would be executed later

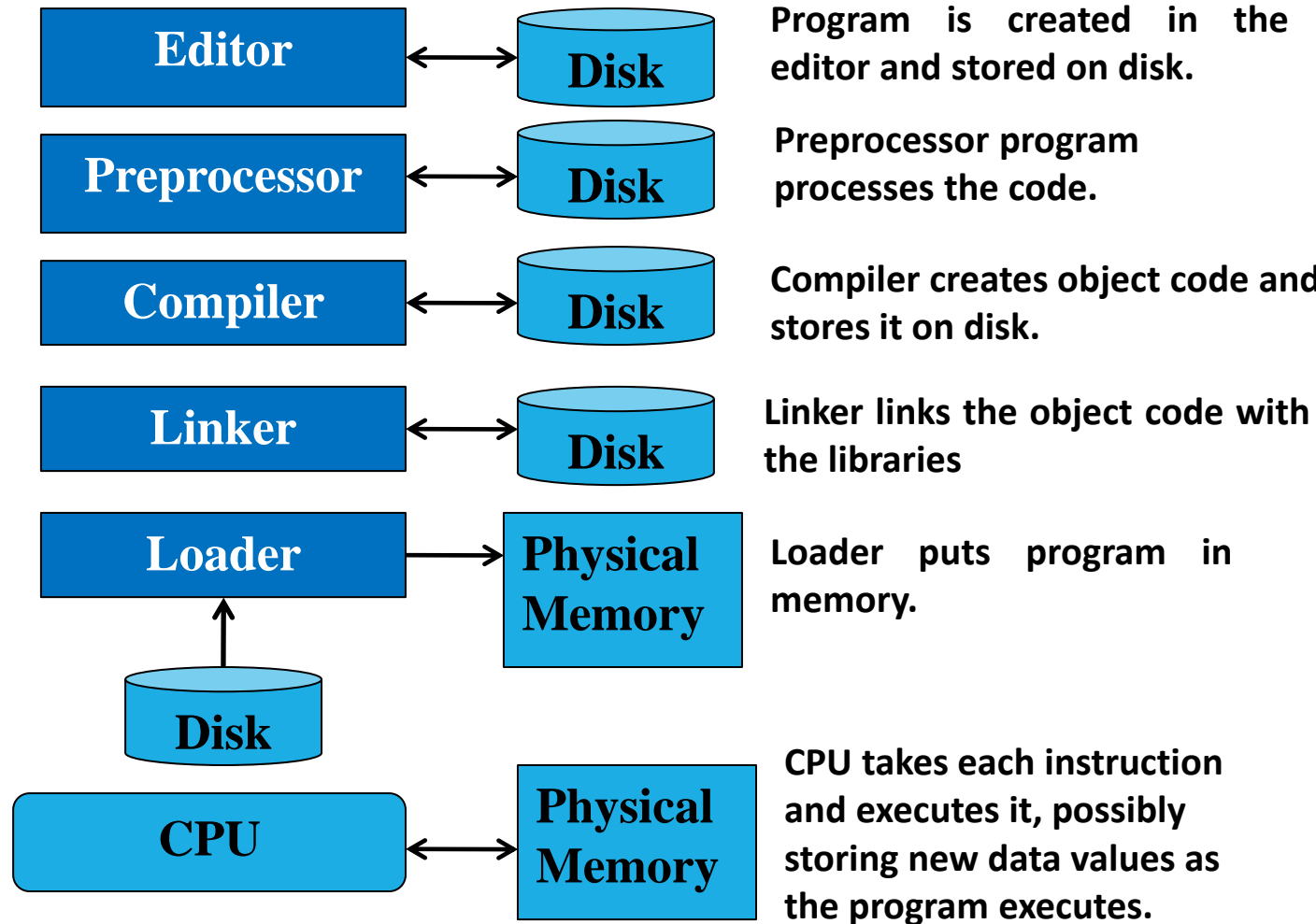
Interpreters

- Execute the instructions as they were translated

C Program Development Environment

Phases of C Programs:

- Edit
- Pre-processor
- Compile
- Link
- Load
- Execute



Natural Languages

Until now we have used the English language to describe the algorithms

Natural languages like English, Russian, Urdu are not completely formal and many times they are vague or ***ambiguous***

The written algorithms used in computers need to be **concise**, **complete** and **non-ambiguous**, therefore, a formal language is used to describe them

Machine Language, Assembly Language and High Level Language

Three types of programming languages

1. Machine languages

- Strings of numbers giving machine specific instructions. Programs written in machine language consist of entirely of 1s and 0s

- Example:

```
00101010 000000000001 000000000010  
10011001 000000000010 000000000011
```

2. Assembly languages

- English-like abbreviations representing elementary computer operations (translated via assemblers)

- Example:

```
LOAD BASEPAY  
ADD OVERPAY  
STORE GROSSPAY
```

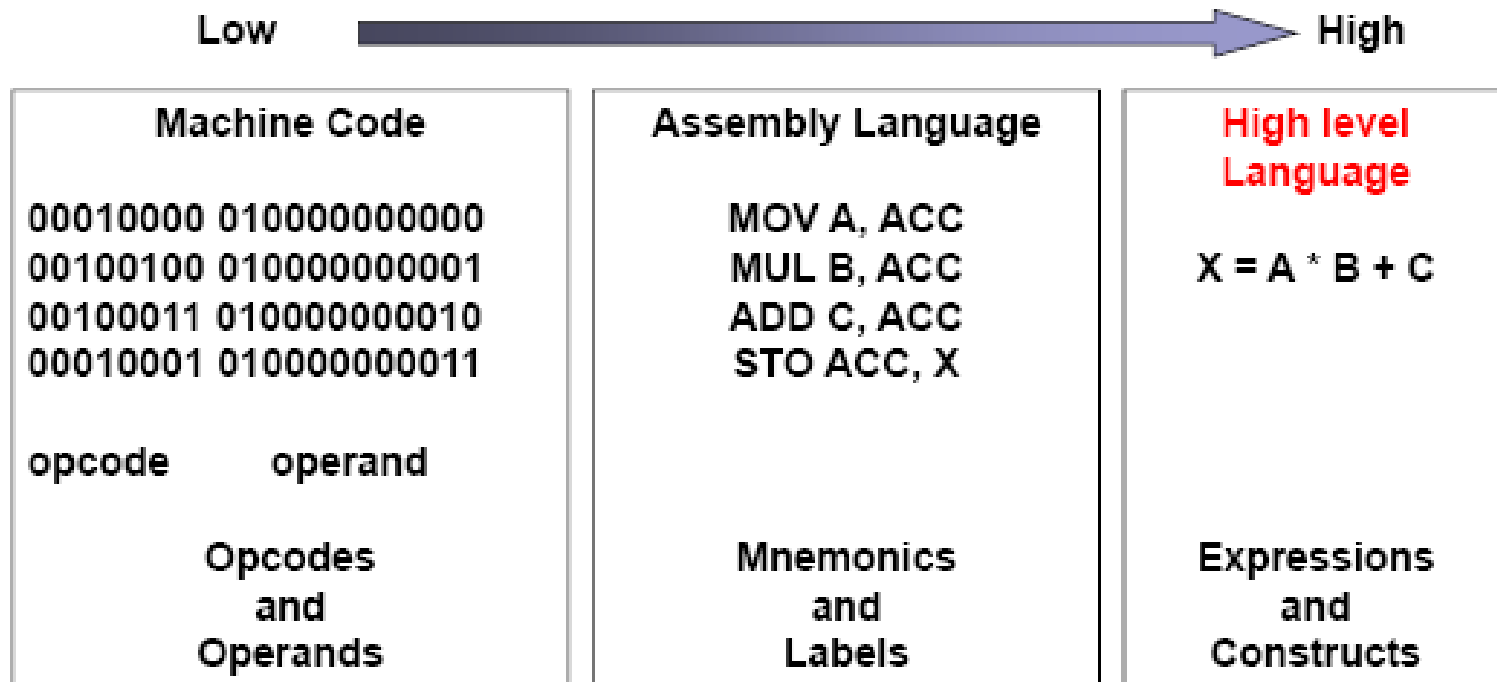
Machine Language, Assembly Language and High Level Language

3. High-level languages

- Codes similar to everyday English
- Use mathematical notations (translated via compilers)
- Example:

`grossPay = basePay + overTimePay`

Programming Levels



History of C

- 'C' language was evolved by Ritchie from two previous programming languages, BCPL and B
- Standard created in 1989, updated in 1999
- C programs consist of pieces/modules called functions
 - A programmer can create his own functions
 - Programmers will often use the C library functions
 - Use these as building blocks
 - Avoid re-inventing the wheel
- C++ → A superset of C was developed by Bjarne Stroustrup at Bell Labs
- Object Oriented Design of C++ - Code REUSABILITY
- Because C++ includes C, it's best to master C then learn C++
- Other object-oriented programming languages are: JAVA, COBOL 2002, FORTRAN 2003 etc

First Program in C

- Writing all the C program now

```
/* This program displays texts on screen */  
#include <stdio.h>  
/* Function main begins program execution */  
int main()  
{  
    printf("Hello World!\n"); /* writes a sentence on the screen */  
    return 0; /* indicates that program ended successfully*/  
}  
/* end function main */
```

Everything written inside `/*` and `*/` are comments, the compiler ignores them completely, but they explain to other programmers what the program does.

Documentation is an import part of programming development, it is an essential good practice in programming !!!