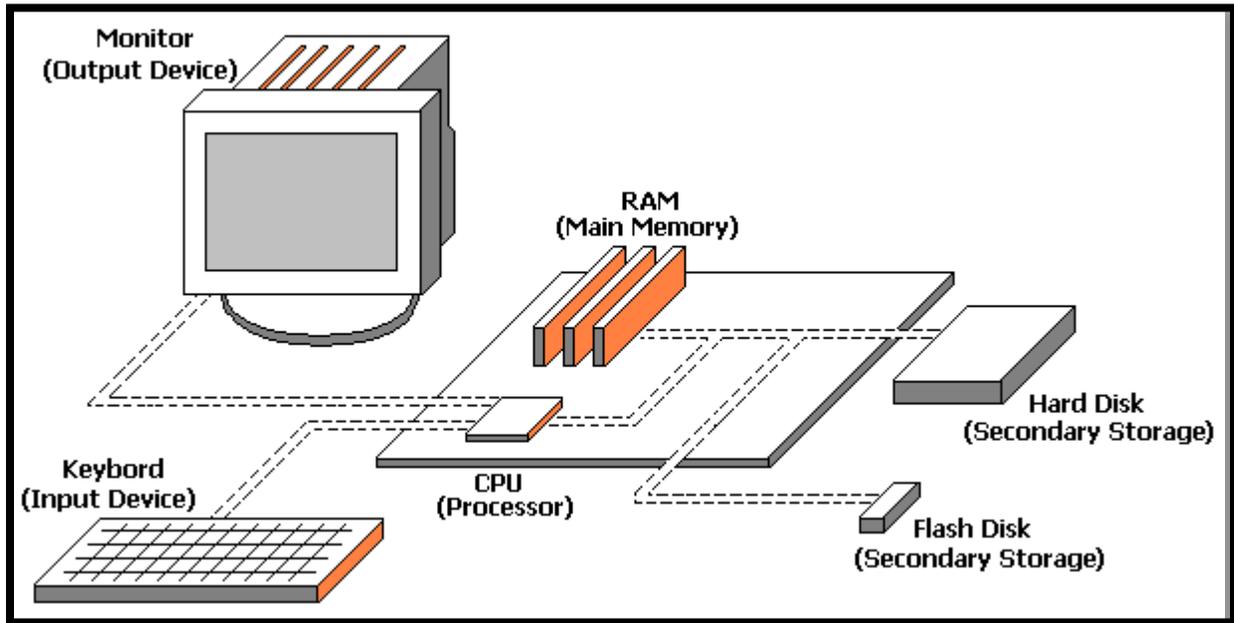# LECTURE 1

1. Introduction:-

Hardware components

Computer: - is a device capable of performing computations and making logical decisions at speeds millions and even billions of times faster than human beings.

Computers process data under the control of sets of instructions called **computer programs**.

Programming is the process of writing instructions for a computer in a certain order to solve a problem.

The computer programs that run on a computer are referred to as **software (SW)**. While the hard component of it is called **hardware** (HW).

2 . C++ Programming Language:

For the last couple of decades, the C programming language has been widely accepted for all applications, and is perhaps the most powerful of structured programming languages. Now, C++ has the status of a structured programming language with object oriented programming (OOP).

C++ has become quite popular due to the following reasons:

1. It supports all features of both structured programming and OOP.

2. C++ focuses on function and class templates for handling data types.

3-**Algorithm:**

As stated earlier an algorithm can be defined as a finite sequence of effect statements to solve a problem. An effective statement is a clear, unambiguous instruction that can be carried out .Thus an algorithm should special the action to be executed and the order in which these actions are to be executed.

Algorithm properties:

➢ The sequence of instructions will terminate.

➢ The instructions are precise. Each instruction is unambiguous and subject to only one interpretation.

➢ The instructions are simple to perform. Each instruction is within the capabilities of the executing agent and can be carried out exactly in a finite amount of time; such instructions are called *effective*.

➢ There are inputs and outputs. An algorithm has one or more outputs (answers) that depend on the particular input data.

Our description of the change-making algorithm, although relatively precise, is not written in any formal programming language. Such informal notations for algorithms are called *pseudocode*, whereas real code is something suitable for a computer. Where appropriate, we use pseudocode to explain an algorithm or computation to you without all the necessary detail needed by a computer.

Examples of Algorithms in Programming

Ex1:-Write an algorithm to add two numbers entered by user.

Step 1: Start

Step 2: Declare variables num1, num2 and sum.

Step 3: input values num1 and num2.

Step 4: Add num1 and num2 and assign the result to sum.

   sum←num1+num2

Step 5: Display sum

Step 6: Stop

Ex2:-Write an algorithm to find the largest among two different numbers entered by user.

Step 1: Start

Step 2: Declare variables a, and b.

Step 3: input variables a, and b.

Step 4: If a>b

  Display a is the largest number.

 Else

  Display b is the largest number.

 Step 5: Stop

EX3: - Write an algorithm to find the area of the rectangle if it given the two dimensions.

The algorithm
1) Start.
2) Declare variables L,W,A.
3) Input L, W.
3) Compute A: A = L*W.
4) Output A.

5) Stop

EX4 :-Write an algorithm to read the number x and then find the value of y from the following equation: $y = 3x^2 - 7x + 8$.

1) Start.

2) Declare variables x ,y.

2) Input x.

3) Compute y: $y = 3 * x^2 - 7 * x - 8$.

4) Output y.

5) STOP.

EX5:- Write an algorithm to find the area and circumference of the circle.

1) Start.

2) Declare variables R, C, A.

2) Input R.

3) Compute A: $A = \pi * R^2$.

4) Compute C: $C = 2 * \pi * R$.

5) Output A, C.

6) STOP.

EX6: - Write an algorithm to convert the temperature degree from the Fahrenheit measurement to the percentage measurement where the relationship between the two measurements

$$C = \frac{5}{9}(F - 32)$$

1) Start.

) Declare variables C, F

2) Input F.

3) Compute C: $C = (5/9)*(F-32)$

4) Output C.

5) STOP.

Ex7: - Write an algorithm to read two numbers and then calculate the sum of square each number and the sum of cubic each number

1) Start.

2) Declare variables A, B, S, X.

2) Input A, B.

3) Compute S: $S = A^2 + B^2$.

4) Compute X: $X = A^3 + B^3$.

5) Output S, X.

6) STOP.
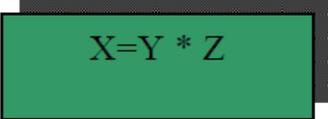
EX8:-Write an algorithm to calculate the value of y from the following relationship

$$y = \begin{cases} x & \text{if} \quad x \geq 0 \\ -x & \text{if} \quad x < 0 \end{cases}$$

1) Start.

2) Declare variables X ,y

2) Input x.

3) If $x = 0$ then $y = x$ else $y = -x$.

4) Output y.

5) stop.

6

The flowchart

A flowchart is a graphical representation of an algorithm or of a portion of an algorithm .Flowcharts are drawn using symbols. The main symbols used to draw a flowchart are shown in following figure.
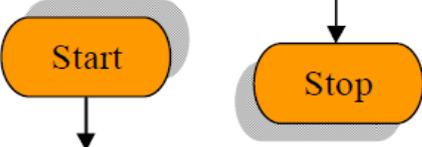
Start Stop	Start and Stop Symbols
Input Output	Input and Output Symbols
X=Y * Z	Mathematical and logical processing symbol
Condition	Decision making symbol
1A 2A	Connector symbols
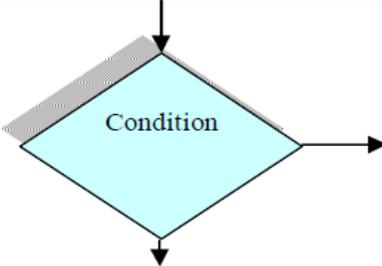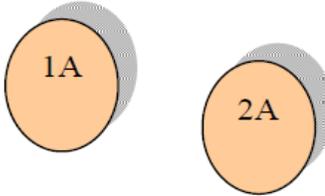
Example 1:

Draw a flowchart to read 3 numbers: x , y and z and print the largest number of them.

```
          ┌─────────────┐
          │    Start    │
          └─────────────┘
                 │
                 ▼
            ╱─────────╲
           ╱  Read x   ╲
           ╲───────────╱
                 │
                 ▼
          ┌─────────────┐
          │    Max=x    │
          └─────────────┘
                 │
                 ▼
            ╱─────────╲
           ╱  Read y   ╲
           ╲───────────╱
                 │           Yes
                 ▼
              ╱─────╲
     No     ╱  y>max ╲
   ┌───────╱    ?     ╲
   │       ╲─────────╱
   │            │
   │            ▼
   │     ┌─────────────┐
   │     │    Max=y    │
   │     └─────────────┘
   │            │
   └────────────┤
                ▼
            ╱─────────╲
           ╱  Read z   ╲
           ╲───────────╱
                 │           Yes
                 ▼
              ╱─────╲
     No     ╱  z>max ╲
   ┌───────╱    ?     ╲
   │       ╲─────────╱
   │            │
   │            ▼
   │     ┌─────────────┐
   │     │    Max=z    │
   │     └─────────────┘
   │            │
   └────────────┤
                ▼
            ╱─────────╲
           ╱ Print max ╲
           ╲───────────╱
                 │
                 ▼
          ┌─────────────┐
          │     End     │
          └─────────────┘
```