Programming what is C++

IN C++ LANGAUGE

Setting By : Lect. Waleed Rasheed

First Lecture

Programming

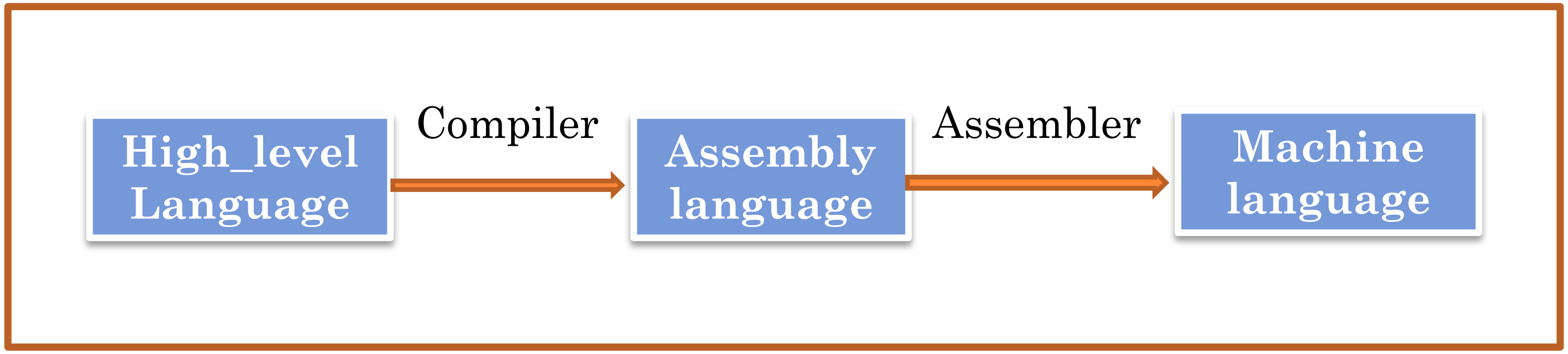
A digital computer is a useful tool for solving a great variety of problems. A solution to a problem is called an algorithm; it describes the sequence of steps to be performed for the problem to be solved.

An **algorithm** is expressed in abstract terms. To be intelligible to a computer, it needs to be expressed in a language understood by it. The only language really understood by a computer is its own machine language. Programs expressed in the machine language are said to be executable. A program written in any other language needs to be first translated to the machine language before it can be executed.

A **machine language** is far too cryptic to be suitable for the direct use of programmers. A further abstraction of this language is the assembly language which provides mnemonic names for the instructions and a more intelligible notation for the data. An **assembly language** program is translated to machine language by a translator called an **assembler**.

Even assembly languages are difficult to work with High-level languages such as C++ provide a much more convenient notation for implementing algorithms.

A program written in a high-level language is translated to assembly language by a translator called a **compiler**. The assembly code produced by the compiler is then assembled to produce an executable program.



Algorithms

Algorithm is a procedure for solving problems. specifies a series of steps that perform a particular computation or task. In order to solve a mathematical or computer problem. An algorithm includes calculations, reasoning and data processing. Algorithms can be presented by natural languages, pseudo code and flowcharts, etc.

Algorithm Properties

1. An algorithm is an unambiguous description that makes clear what has to be implemented.
2. A possible way to solve a given problem
3. A mandatory first step before implementing a solution
4. An algorithm expects a defined set of inputs.
5. An algorithm produces a defined set of outputs.
6. An algorithm is guaranteed to terminate and produce a result, always stopping after a finite time. If an algorithm could potentially run forever, it wouldn't be very useful because you might never get an answer.

Creating an Algorithm

To begin with we will look at three methods used in creating an algorithm, these are :

STEPPING, LOOPING, CHOOSING

STEPPING: This is where all the instructions needed to solve our problem are set out one after the other. Here are some examples:

PROBLEM: To find the sum of two numbers.

ALGORITHM:

1. add the two numbers together

2. write down the answer.

Creating an Algorithm

ALGORITHM:

1. Multiply the two numbers
2. write down the answer.

Example : Write an algorithm and draw a flowchart that will read the two sides of a rectangle and calculate its area.

```
Input: Width, Length
Output: Area Algorithm
Step 1: Input W, L
Step 2: Area = L x W
Step 3: Print Area
```

GOING LOOPING

if we have to repeat an instruction or a set of instructions a number of times to find a solution for specific problem? In this case we can use loops. We will look at three different types of loops:-

- the REPEAT UNTIL loop
- the WHILE loop
- the FOR loop

PROBLEM: To print the numbers from 10 to 20

ALGORITHM:

```
1.For number = 10 to 20
2.Print number
3.END FOR
```

CHOOSING [IF, THEN and ELSE]

we will look at a method for making a choice or a decision. This technique is called CHOOSING, and uses the commands IF, THEN and sometimes (but not always) ELSE.

With this type of command a condition is given with the IF command followed by an action to be taken. If the condition is satisfied, we follow it by the THEN command. A couple of examples which should make things a little clearer:-

CHOOSING [IF, THEN and ELSE]

Example: Write an algorithm to determine a student's final grade and indicate whether it is passing or failing. The final grade is calculated as the average of four marks.

Input: M1, M2, M3, M4

Output: Grade

Algorithm

```
Step 1: Input M1,M2,M3,M4
Step 2: GRADE = (M1+M2+M3+M4)/4
Step 3: If (GRADE < 50) then
            Print "FAIL"
else
            Print "PASS"
End If
```

CHOOSING [IF, THEN and ELSE]

Example: Problem: Given a list of positive numbers, return the largest number on the list.
Inputs: A list L of positive numbers. (This list must contain at least one number).
Outputs: A number n, which will be the largest number of the list.
Algorithm:

```
1.Set max to 0.
2.For each number x in the list L,
3.If x is larger than max, then set max to x.
4.End For
5.Print max is now set to the largest number in the list.
```

CHOOSING [IF, THEN and ELSE]

Exercises

Determine and Output Whether Number N is Even or Odd

Algorithm:

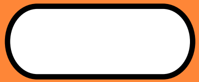
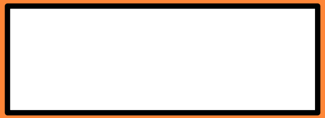
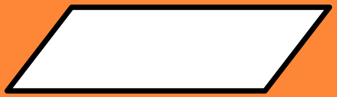
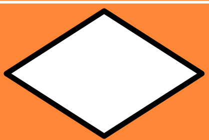
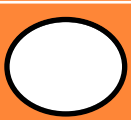
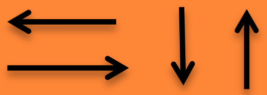
```
1. Read number N,
2. Set remainder as N modulo 2,
3. If remainder is equal to 0 then
       number N is even,
       else number N is odd,
4. End If
```

FLOWCHART

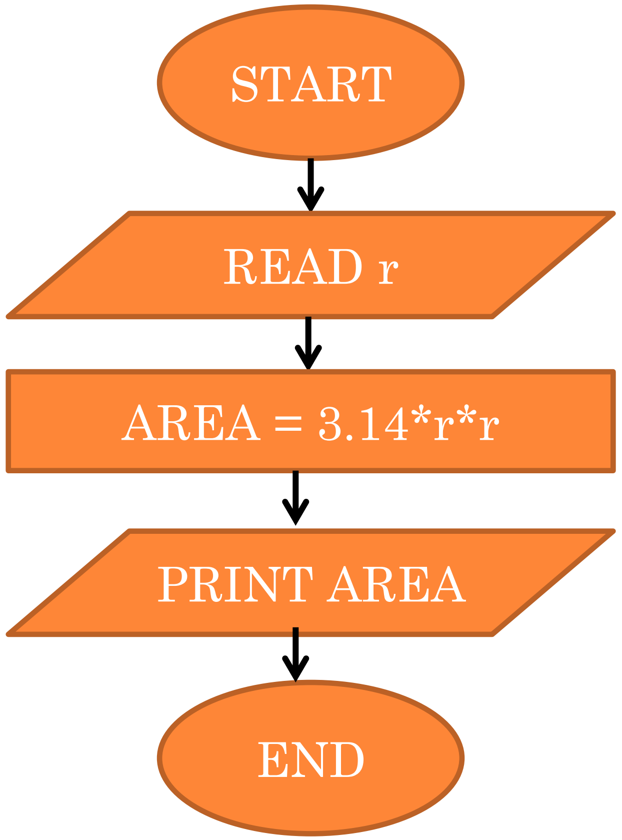
The flowchart is a diagram which visually presents the flow of data through processing systems. This means by seeing a flowchart one can know the operations performed and the sequence of these operations in a system. Algorithms are nothing but sequence of steps for solving problems. So a flow chart can be used for representing an algorithm.

Flowchart Symbols

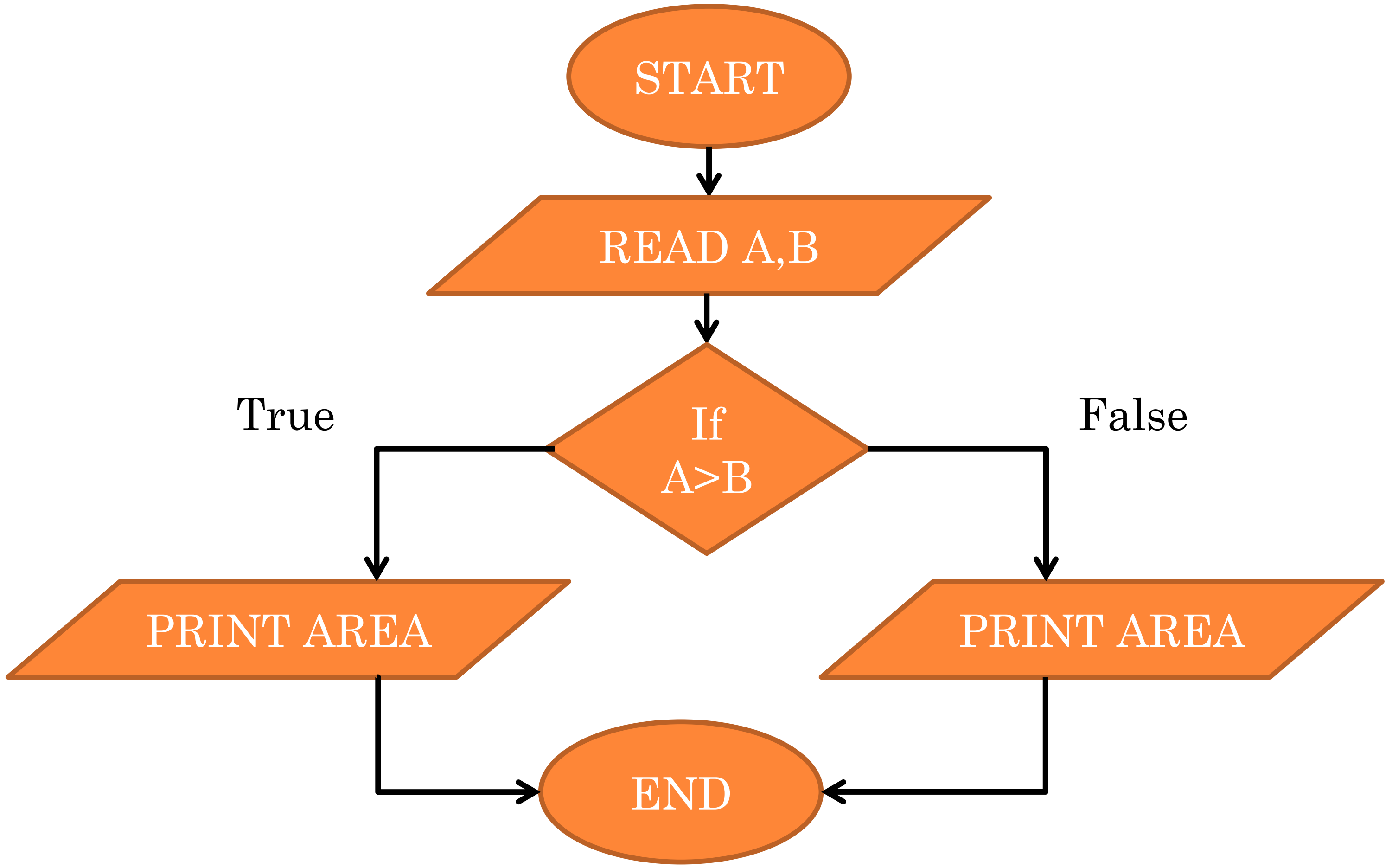
The basic symbols commonly used in flowchart drawing in Programs are: Process, input/output, Decision, Connector and Flow Lines, described as follows:

Symbol	Function
	starting or ending of the program
	Indicates any type of internal operation inside the Processor or Memory
	Used for any Input / Output (I/O) operation. Indicates that the computer is to obtain data or output results.
	Used to ask a question that can be answered in a binary format (Yes/No, True/False)
	Used for connection,
	Shows direction of flow.

Example: draw a flowchart to Find the area of a circle of radius r.



Example: Draw a flowchart to find the greater number between two numbers .



What is C++?

C++ is a general purpose programming language with a bias towards systems programming that
– is a better C,
– supports data abstraction,
– supports object oriented programming, and
– supports generic programming.

Object Oriented Programming is a technique for programming – a paradigm for writing "good" programs for a set of problems. If the term 'object oriented programming language" means anything, it must mean a programming language that provides mechanisms that support the object oriented style of programming well.

Procedural Programming
The original programming paradigm is: (Decide which procedures you want; use the best algorithms you can find). The focus is on the processing – the algorithm needed to perform the desired computation. Languages support this paradigm by providing facilities for passing arguments to functions and returning values from functions.

A Simple C++ Program

first C++ program, which when run, simply outputs the message Hello World.

```
#include <iostream>
using namespace std;
int main()
{
   cout << "Hello World\n";
   return 0;
}
```

#include : to include the contents of the header file iostream in the program. iostream is a standard C++ header file and contains definitions for input and output.

{ : This brace marks the beginning of the body of main.
} : This brace marks the end of the body of main.

A Simple C++ Program

int main ( )This line initiates the declaration of a function. Essentially, a function is a group of code statements which are given a name: in this case, this gives the name "main" to the group of code statements that follow. All C++ programs must have exactly one main function. Program execution always begins from main.

std::cout: which identifies the **st**andar**d c**haracter **out**put device

using namespace std; to avoid write code like "std::cout"

We could have written:

```
int main () { std::cout << "Hello World!"; }
```

A Simple C++ Program

cout << "Hello World\n"; A statement is a computation step which may produce a value. The end of a statement is always marked with a semicolon (;). This statement causes the string "Hello World\n" to be sent to the cout output stream. A string is any sequence of characters enclosed in double-quotes. The last character in this string (\n) is a newline character which is similar to a carriage return on a type writer. A stream is an object which performs input or output. cout is the standard output stream in C++ (standard output usually means your computer monitor screen). The symbol << is an output operator which takes an output stream as its left operand and an expression as its right operand, and causes the value of the latter to be sent to the former. In this case, the effect is that the string "Hello World\n" is sent to cout, causing it to be printed on the computer monitor screen.