Tikrit University

Petroleum and Minerals College

Control and Computer Engineering Department

C++ Programming

Introduction

1st Year

1. Introduction to Computer and Programming

Computer is now an essential part of our daily life, and also an important factor in science and technology.

A computer system consists of hardware, the mechanical and electronic devices that are capable of computing and manipulating information, and software, i.e., programs that carry out predefined instruction to complete a given task. Computer software communicates with the hardware to perform useful tasks – the software organizes the control sequences, and the hardware carry out the instructions defined by the software.

1.1 Electronic Computers

The advance of semiconductor technology has made computing affordable.

The reduction in price and improvement in performance has made computer much more affordable and powerful.
The first important improvement of so-called computer is from mechanical to electronic in 1930s.

The first large-scale digital computer ENIVAC was built in 1946 by University of Pennsylvania.

- **The first generation computer**, built between 1939 and 1957, used vacuum tubes, which consumed a lot of power and space. They tend to overheat and burn out quickly; as a result they were expensive and unreliable. They are mainly for scientific use due to its high costs.
- **The second-generation computers** use transistors instead of vacuum tubes. They use less power, take up less space, and become more reliable and less expensive. Now the computers become more affordable and more business can use them.
- **The third-generation computers** use integrated circuit (IC)

technology that integrates many electronic components into a single silicon chip, to make computers faster, smaller, and cheaper.

- **The fourth generation computers** take one step further and introduce Large-scale integration (LSI) and Very Large Scale Integration (VLSI) so that we can package an entire processor into a single chip.

Programming has evolved from a tedious error-prone job to machine-assisted, semi-automatic process.

Computers are classified according to their sizes and speed. They are:

- ***Microcomputer.***
- ***Minicomputers.***
- ***Mainframe computers.***
- ***Super computers.***

1.2 Components of a Computer

There are four basic components in a computer system hardware.

- *Central Processing Unit (CPU)*

  The main purpose of CPU is to process data in main memory. CPU is capable of arithmetic and logic operation on data stored in main memory.

- *Main memory*

  The computer software (or program) and the data it wants to manipulate are stored in main memory. The instruction of program are fetched from memory by CPU and executed. The data are also stored in memory to retain their value, or state.

- *Secondary storage*

  The secondary storage provides large, non-volatile, and inexpensive storage for program and data.

- *Input/Output (I/O) devices*

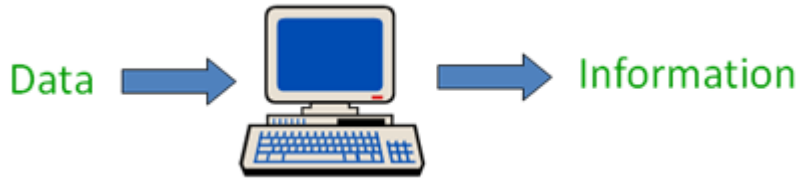
  I/O devices provide channels to transmit/receive data to/from a computer.

Memory consists of cells, which are indexed by address. A cell consists of several bytes. Each byte has 8 bits. Memory can be random access memory (RAM) or read-only memory (ROM), or FLASH. RAM can only store data temporarily, i.e. it is a volatile storage device.

Secondary storage provides semi-permanent storing capability. Hard disk is the most popular secondary storage devices for its high capacity and low costs.

CPU is a complicated chip to handle all the computation within a computer. It has registers and cache memory to store the instruction and data retrieved from main memory, and complicated decoding circuit to determine what action should be performed, and an ALU (arithmetic and logic unit) with circuit to carry out the actual computation.

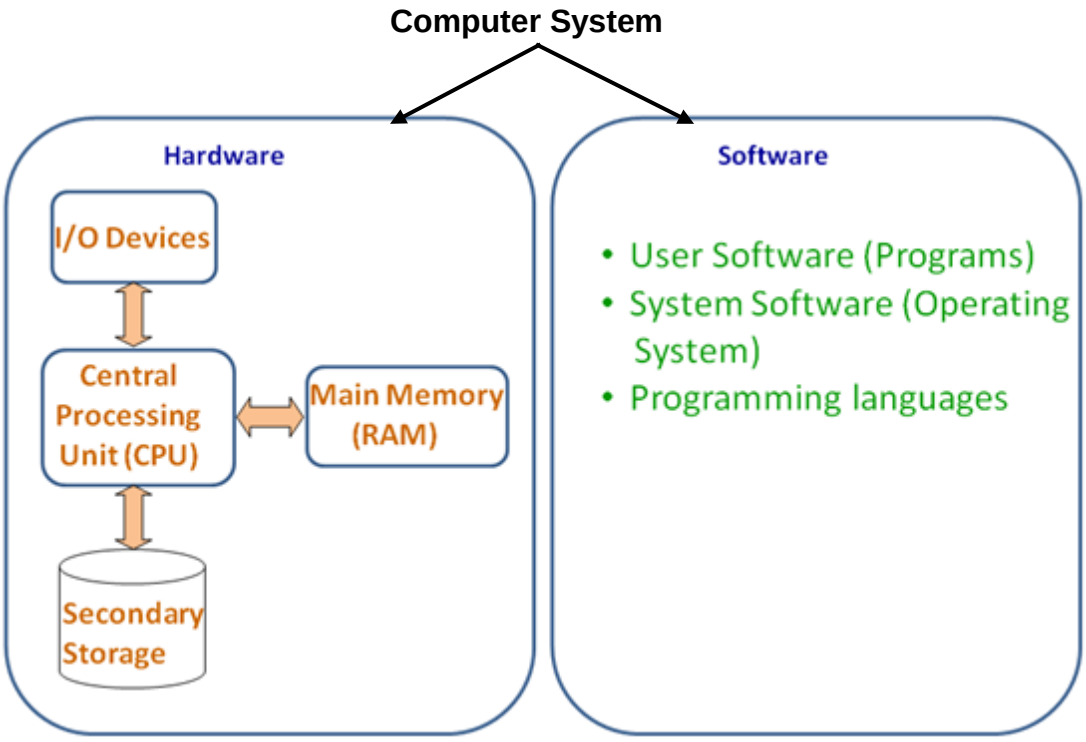
A computer is a machine that is designed to perform operations that are specified with a set of instructions called a program. It can perform computations and make logical decisions faster than human beings can.



Information could be:

- Results of scientific calculations
- Results of statistical analysis
- Conclusions of logic deduction
- Results of information retrieval or data summary

A **computer system** consists of 1) **Hardware,** 2) **Software and 3) User.**



A **program** is a sequence of instructions that instructs the computer what to do.

What is programming?

- Programming is the ability to get computer to perform useful tasks for us
- Programming is about problem solving
- Programming is about solution development
  - From the problem to an algorithm
  - From algorithm to a program
- Programming is about logic
- Programming is about how computer works

1.3 Programming Languages

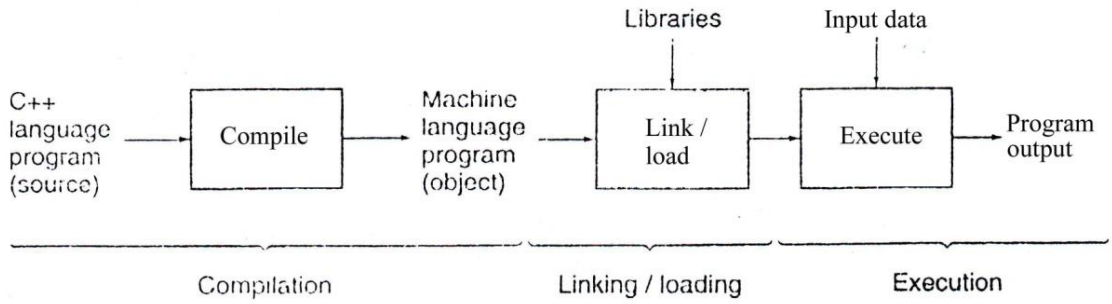
- Types of Programming Languages
  - Machine language
  - Assembly language
  - High level languages

- High Level Programming Languages
  - Imperative (Procedural) Languages e.g. Bascal and C
  - Object-Oriented Languages e.g. C++ and Java
  - Other Types of Languages e.g. Prolog
- Programming Style
  - Structured (procedural) programming
  - Object-oriented programming
  - Event-driven programming

1.4 C++ Program Execution

- ***Editing***: writing program source code
- ***Compiling:*** finding syntax errors, if errors are found then debug, else generate object program (program in machine language)
- ***Debugging:*** correcting compile-time errors , then recompile
- ***Linking:*** connecting to existing utilities (e.g. Libraries)
- ***Loading:*** loading the program into memory
- ***Running:*** execute (run) the program, if run-time errors are found then correct them and recompile.

The processes of compilation, linking/loading, and execution are outlined in the figure below.



1.5 Problem-Solving Methodology

The process for problem-solving has five steps:

1. State the problem clearly.
2. Describe the input and output information.
3. Work the problem by hand (or with a calculator) for a simple set of data.
4. Develop an algorithm and convert it to a computer program. The algorithm can be listed as operations that are performed one after another.
5. Test the program with a variety of data.

1.6 Development of Algorithms

To describe the process of solving a problem with sequential logic, we use the **top-down design**. There are several ways to develop an algorithm:

- *Decomposition outline:* is written sequentially and represents an ordered set of steps.

  Example:

  Compute the straight-line distance between two points in a plane.

  Decomposition outline:

  1. Give values to the two points.
  2. Compute the lengths of the two sides of the right triangle generated by the two points.
  3. Compute the distance between the two points, which is equal to the length of the hypotenuse of the triangle.
  4. Print the distance between the two points.

- *Pseudocode:* uses English-like statements to describe the steps in an algorithm.
- *Flowcharts:* uses a diagram to do the same.

The next example shows an algorithm described by a pseudocode and flowchart.

Example:

Pseudocode Notation **Flowchart Symbol**

main:

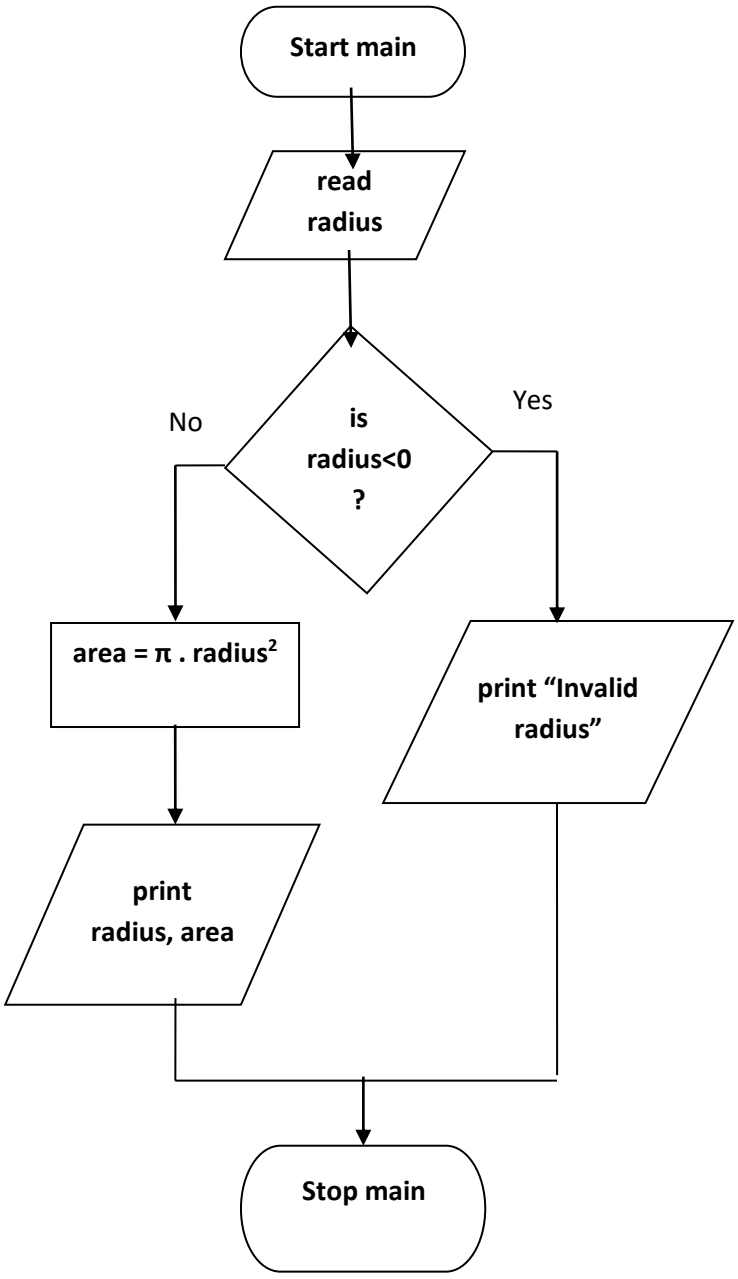
read radius

if radius < 0 then
print “Invalid radius”
else

set area to π . radius^2

print radius, area

end



1.7 C++ Program Structure

- Comments
- Libraries
- main function
  - Statements that define memory locations (constants and variables)
  - Statements that specify actions to be taken (i.e. flow of controls)
  - Operational expressions involving data representations
- Representations of data of different types (e.g. structure)
- Other Functions and procedures

```
// Comments
.....
.....
// Comments

Links to Libraries
...... Lib 1
......
...... Lib M
Links to Libraries

Main Program
{
  input & initialisation
  Expressions, Statements
  function calls
  output & termination
}
```

A Simple C++ Program

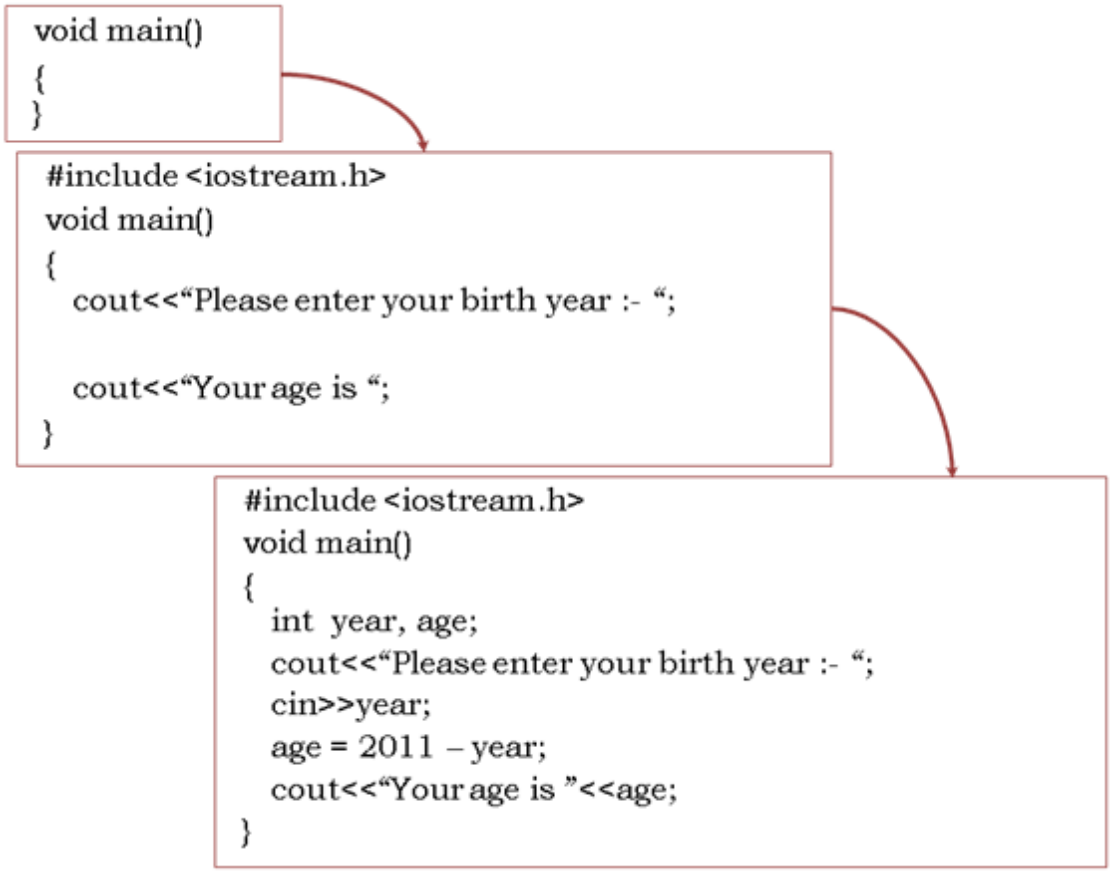
```
// welcome.cpp
//
// This program prints
// a welcome message

#include <iostream.h>

void main()
{
  // print welcome message
  cout<<“Welcome to C++ programming“;
}
```

1.8 Principle of Step-wise Refinement

A program is developed via a sequence of passes and through a series of versions. Each version is a further refinement of the previous one. Each version must be able to compile and run correctly.

Note:

- Always have a working version
- Identify errors as they appear
- Most time effective



References:

1. Delores M. Etter, "Introduction to C++ for Engineers and Scientists", 1997, Prentice Hall.
2. Robert Lafore, "Object-Oriented Programming in C++", 4th edition, 2002, SAMS publishing.
3. H. M. Deitel and P.J. Deitel, "C++ How to program", 8th edition, 2012, Prentice Hall.
4. E Balagurusamy, "Object-Oriented Programming with C++", 7th edition, 1997, McGraw-Hill.
5. H. M. Deitel and P.J. Deitel, "Java How to program", 7th edition, 2006, Prentice Hall.
6. Herbert Schildt, "Java: The Complete Reference", 7th edition, 2007, Mc-Graw-Hill.