Ministry of Higher Education
Karwan University
Faculty of Computer Science

Programming I

Lecture 1

Lectures & other related materials are available here: https://github.com/mujtabaSultani01/Programming-I

Contents

- Programming Languages
- POL vs OOP
- Brief history of C/C++ Programming Languages
- C++ Standards
- Steps throughout Code Execution
- C++ Syntax
- Writing your first C++ program
- Discussion

What is programming language?

Programming Languages

- Machine Language
- Assembly Language
- High Level Language

Machine Language

- The **fundamental language** of the computer's processor, also called **Low Level Language**.
- ***Machine language*** is a set of built-in primitive instructions. These instructions are in the form of **binary code**, so if you want to give a computer an instruction in its native language, you have to enter the instruction as binary code.
- **For example**, to add two numbers, you might have to write an instruction in binary code, as follows:

1101101010011010

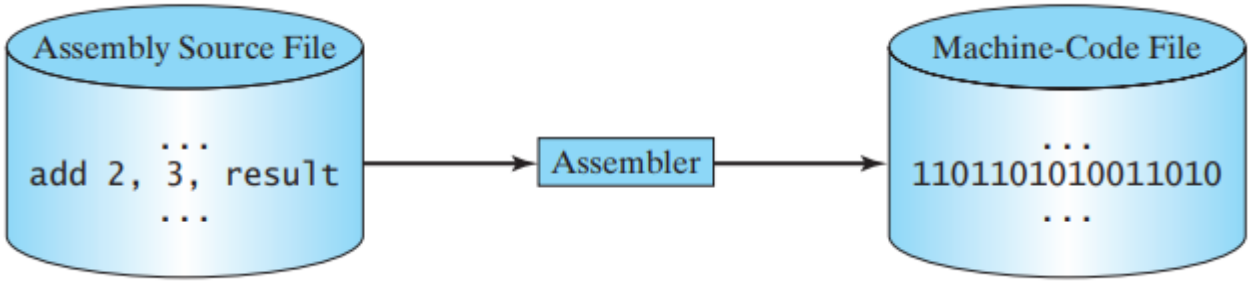
Assembly Language

- Programming in machine language is a tedious process.
- Moreover, programs written in machine language are very difficult to **read** and **modify**. **For this reason, assembly language** was created in the early days of computing.
- Uses **symbolic operation code** to represent the machine operation code.
- **For example**, to add the numbers 2 and 3 and get the result, you might write an instruction in assembly code like this:

add 2, 3, result

Assembly Language

- Assembly languages were developed to make **programming easier**.
- However, because the computer cannot execute assembly language, another program called an **assembler** is used to translate **assembly-language programs into machine code**.



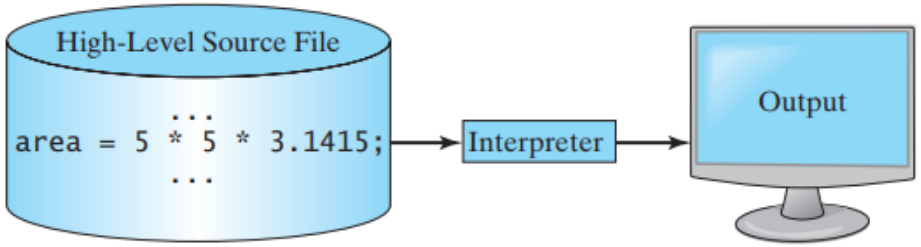
High-Level Languages

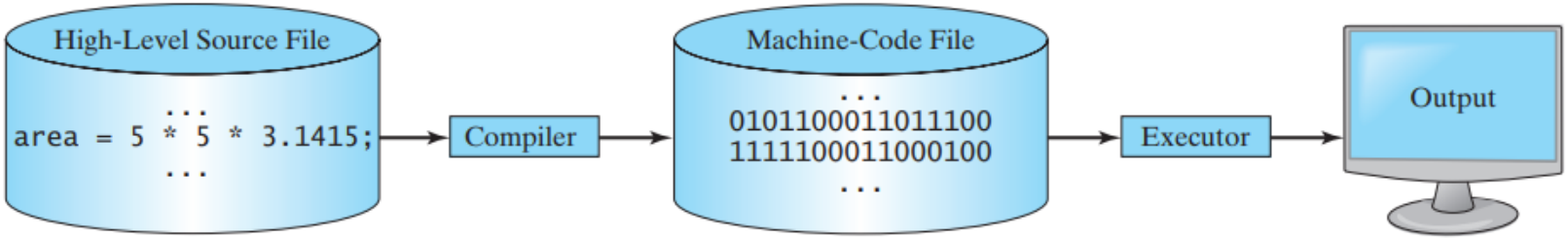
- In the **1950s**, a new generation of programming languages known as **high-level languages** emerged.
- They are **platform-independent**, which means that you can write a program in a high level language and run it in different types of machines.
- The instructions in a high-level programming language are called **statements**.
- Computer (programming) languages that are easier to learn.
- Examples are C ++, Python, Pascal, Fortran, R and . . .

area = 5 * 5 * 3.1415

High-Level Languages

- **Interpreted** Languages
- **Compiled** Languages





High-Level Languages

- Procedural Oriented Programming Language (**POP**)
- Object Oriented Programing Language (**OOP**)

Procedural Programming Languages

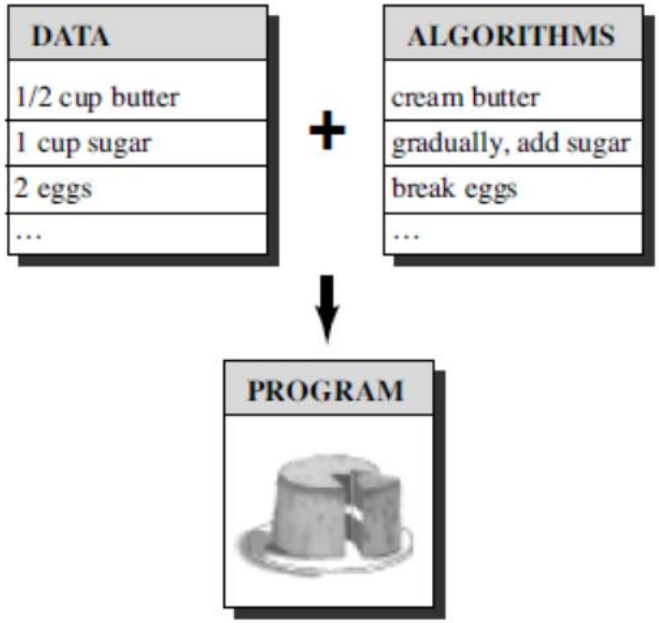
- **Procedural programming** uses a list of instructions to tell the computer what to do **step-by-step**.
- It contains a systematic order of statements, functions and commands to complete a computational task or program.
  - ✓ Uses Top Down Approach
  - ✓ Block branching while, for and switch statements

Object-oriented Programming Languages

- Object-oriented programming, or OOP, is an approach to problem-solving where all computations are carried out using **objects**.
- **An object** is a component of a program that knows how to **perform certain actions** and how to **interact** with other elements of the program.
  - ✓ Bottom up Approach
  - ✓ Objects like class

C Programming Language

- The origin of C is closely tied to the development of the Unix operating system.
- Invented by **Dennis Ritchie** in 1972 AT&T Bell Labs.
- C is procedural or Algorithmic Programming language.



C Programming Language

- C++ is an object oriented programming language.
- **Bjarne Stroustrup**, a Danish computer scientist, began his work on C++'s predecessor "C with Classes" in 1979. He was researching for his thesis while faced difficulties analyzing Linux Kernel.
- Bjarne Stroustrup enhanced C with adding more features to it.
- In 1983, it was renamed from C with Classes to C++ (++ being the increment operator in C).

Benefit of Using OOP over Structural Programming Language

- **OOP** emphasizes on **problem** instead **blocks of code.**
- OOP **simulates** real world objects uses class.
- Uses **bottom-up** approach.
- OOP facilitates creating **reusable code.**
- Instead of **concentrating** on tasks you concentrate on representing concepts.

Portability & Standards

- C++ Programming Language is not **machine defended**.
- C++ code can be recompiled on various **vendor**.
- C programs can be run with C++ compiler.

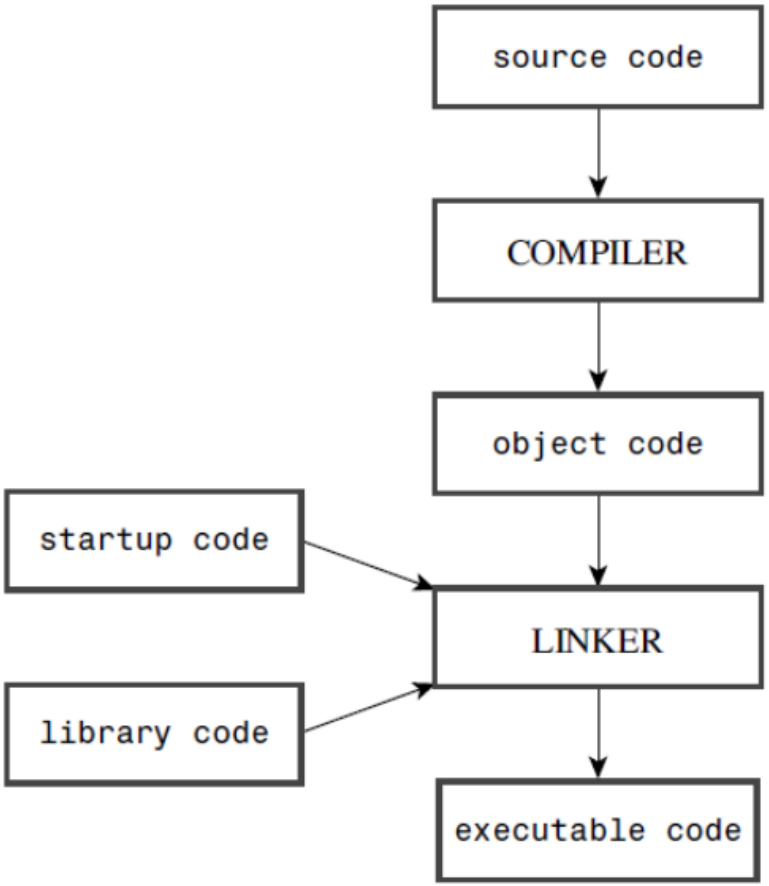
C++ Standards

- Different accent in a language can arise many problems in understanding the language. Programming language has no acception. If different developers develops their own version of C++ this can make the language hard to understand.
- American National Standards Institute (ANSI) created a committee in 1990 (ANSI X3J16) to develop a standard for C++.
- International Organization soon joined the committee.
- C++98
- C++03
- C++07/TR1
- C++11
- C++14
- C++17

Steps throughout Code Execution

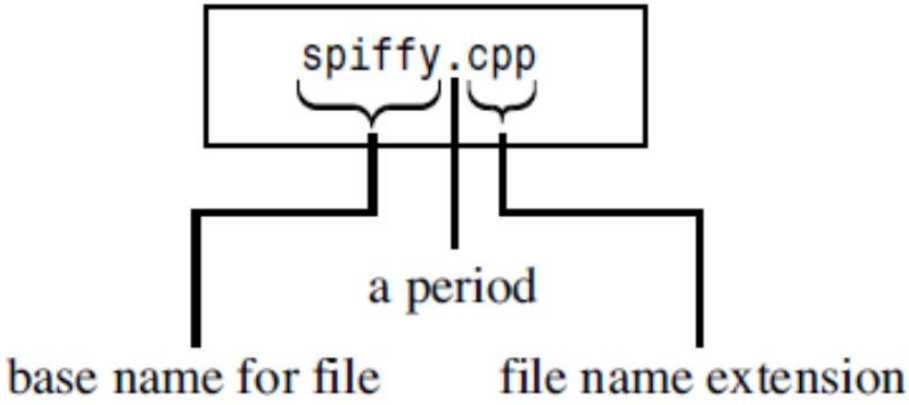
- Suppose you've written a C++ program. How do you get it running?
- You need a text editor to write **source code**.
- Compiling the source code to machine language with.....
- After compilation **object code** is generated.
- Object code is then link to C++ integrated libraries to produce the run time version of the program called **executable code.**

Steps throughout Code Execution



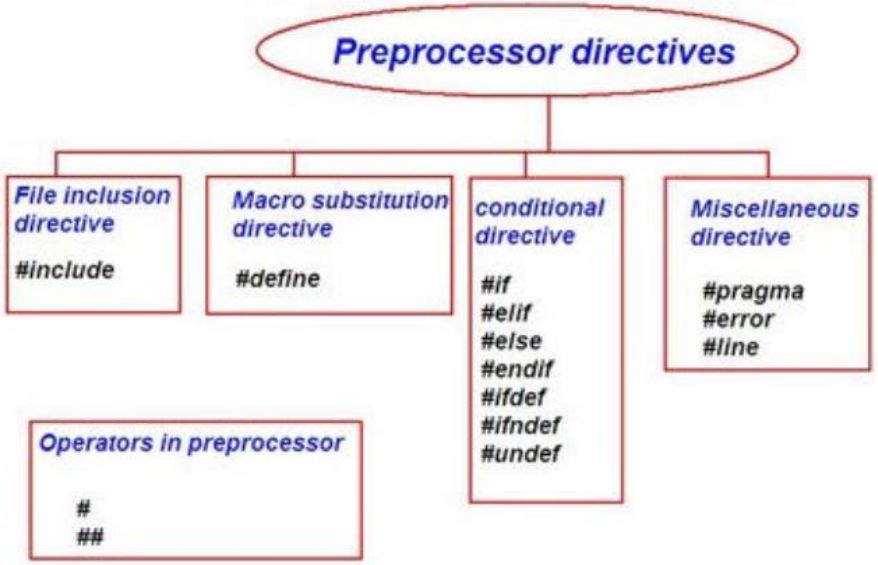
Writing Source Code

- Source code can be written in any text editor (notepad, nano, vi).
- Integrated Development Environments (IDEs).
- write code in any editor save it with proper C++ extension and compile it with appropriate compiler.



Preprocessor Directory

- **Preprocessor directives** are lines included in the code of programs preceded by a hash sign (#) followed by the directory like **inlclude.**
- These directories are used at the **beginning of the source** code to instruct compiler what to do before the actual compilation.



Simple Hello World Program

C:\Users\ADMIN\Desktop\dev c\Hello World.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Profiling

(globals)

Hello World.cpp

```
1 #include<iostream>
2 using namespace std;
3 int main()
4 {
5 cout<<" Hello world !!! ";
6 return 0;
7 }
```

Compiler Resources Compile Log Debug Find Results

Line: 1 Col: 1 Sel: 0 Lines: 7 Length: 102 Insert Done parsing in 1.235 seconds

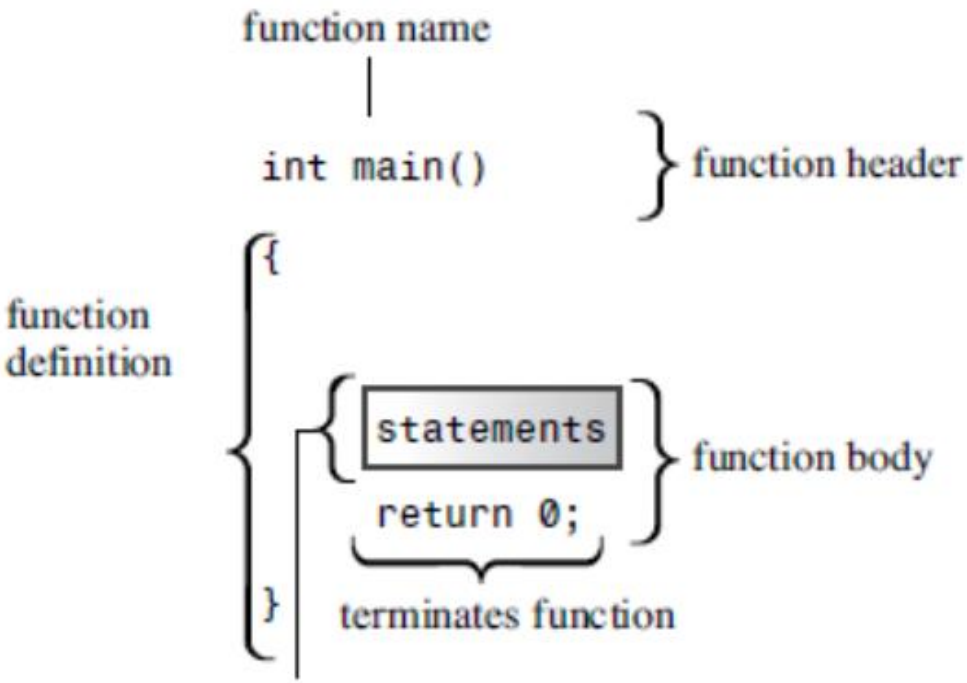
Type here to search ENG 10:59 13-08-2019

C++ Syntax

- Every C++ program should have header file iostream.
- iostream is used for inputting data and outputting information in C++.
- Main function is important for running C++ program.
- main function ends with return 0 finally closed with brace.
- Semicolon(;) is used to end a statement.

Features of the Main() Function

- Main function definition has two parts.
- First function header **int main().**
- Second function **body included in {}.**



Features of the Main() Function

- **Cout** is used to **output** something to the **screen** instead of printf in C.
- **Cin** is used to take an **input** from the **keyboard** instead of scanf in C.

Home Work

- Install C++ Compiler (Dev++, …)
- Write your first 'Hello World' C++ Program.

References

- Starting out with C++ Early Objects.

Questions…?