FAST National University of Computer and Emerging Sciences Peshawar

OOP Lab # 1.1

Introduction to C++

Instructor: Muhammad Abdullah Orakzai

DEPARTMENT OF COMPUTER SCIENCE

Prepared By: Muhammad Abdullah Orakzai (Instructor CS)

Contents

Computer Program

- A computer program is a collection of instructions that performs a specific task when executed by a computer.
- Most computer devices require programs to function properly.
- A computer program is usually written by a computer programmer in a programming language.

Computer Programming

- Computer programming is the process of designing and building an executable computer program to accomplish a specific computing result.
- Used to solve given problems.
- Programming is the process or procedure through which we create and prepare software.

Software

- Set of programs that perform specific task.

Language (General Definition)

- A way or mode of communication.
- It may be verbal or non verbal.
- verbal means spoken or written form.
- Non verbal means non spoken **e.g.** gesture or symbols.

Language (Building Blocks)

All languages have 3 main building blocks

1) Symbols means sign or notations
2) Syntax means the way of writing symbols or the arrangement of symbols or words.
3) Semantic means meanings

Computer Language

- The source of communications.
- A Computer language includes various languages that are used to communicate with a Computer machine.
- Some of the languages like programming language which is a set of codes or instructions used for communicating the machine.
- Machine code is also considered as a computer language that can be used for programming.

Programming Language

- A programming language is a formal language, which comprises a set of instructions that produce various kinds of output.
- Programming languages are used in computer programming to implement algorithms.

C++ Introduction

- C++ pronounced as “See Plus Plus”.
- It is a powerful computer programing language.
- It is the advance version of C language.
- The C is a procedural programing language while C++ is an Object Oriented Programing language.
- Procedural means step by step process or step by step order.

C++ History

- In 1972 Dennis Ritche developed C language at Bell Laboratory.
- C language was an advanced version of B language.
- In 1980 C++ language was developed by “Bjarne Stroustrup” (بجارنی اسٹراسٹرپ).
- C++ is an extension of C.

Writing a C++ program

- Extension of C++ program is cpp i.e. filename.cpp
- Text editor is required to write and edit C++ source code.
- C++ Compiler is required for compilation of source code into machine code. This machine code is called object code. It is stored in a new file with extension ***obj***. The object code is then linked to the libraries. After linking the object code to the libraries, an executable file with extension ***exe*** is created. The executable program is then run from operating system command line.

Writing a C++ program…

- For example a source code of program in C++ is written and stored with first.cpp.
- After compilation, the object code is saved in file first.obj and executable code is stored in file first.exe.
- Before compilation code is called **source code.**
- After compilation code is called **object code.**

Structure C++ program

- A C++ program consists of three main parts. These are:

1. Preprocessor directives
2. The main() function
3. C++ statements

1) Preprocessor directives

- Provides instructions to the compiler before the beginning of the actual program. Also called **compiler directives**.
- The preprocessor directives consist of instructions for the compiler.
- The compiler adds special instructions or code from these directives into program at the time of compilation.

1) Preprocessor directives…

- The preprocessor directives normally start with a number sign **(#)** and the keyword “include” or “define”. For example preprocessor directives are used to include header files in the program.
- A program example is given below. The first statement of the program is a preprocessor directive. The preprocessor directive has been written to include the iostream.h header file.

1) Preprocessor directives…

```
#include <iostream>
int main()
 {
   cout << "Hello C++ Programming";
   return 0;
  }
```

Header File

- Header file is a C++ source file that contains definitions of library functions/objects.
- A header files are added into the program at the compilation of the program.
- A header file is added if the function/object defined in it is to be used the program.
- The preprocessor directive “**include**” is used to add a header file into the program. The name of the file is written in single brackets (<>) after “**include**” directive.

Header File…

- The C++ has a large number of header files in which library functions are defined.
- The header file must be included in the program before calling its function in the program.
- A single header file may contain a large number of built-in library functions.

Header File…

For example the header file iostream.h has definitions of different built in input and output objects and functions. It is included in the above program because its object “cout” is used in the program.

Syntax is given by

#include<name of the header file>

Header File…

Eg:

#include<iostream.h> **--->** input and output stream

#include<conio.h> **--->** console input and output (console means screen)

For example cout<< is defined in iostream.h and getch() is defined in conio.h.

2) The main() function

- The main() function indicates the beginning of C++ program.
- When we run a program first of all OS runs main function.
- Main function is built in function.
- It is automatically called.
- Without it programs is not executed.

1) The main() function…

Syntax

```
int main()
{
program statements…
}
```

3) C++ statements

- The statements of the program are written under the main() function between the curly braces {}.
- These statements are the body of the program.
- Each statement in C++ ends with semicolon (;).
- C++ is case sensitive language. The C++ statements are normally written in lowercase letters but in some exceptional cases, these can also be written in upper case.

C++ Program Syntax

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

Example explained

Line 1: #include <iostream> is a **header file library** that lets us work with input and output objects, such as cout (used in line 5). Header files add functionality to C++ programs.

Line 2: using namespace std means that we can use names for objects and variables from the standard library.

Don't worry if you don't understand how #include <iostream> and using namespace std works. Just think of it as something that (almost) always appears in your program.

Line 3: A blank line. C++ ignores white space.

Example explained…

Line 4: Another thing that always appear in a C++ program, is int main(). This is called a **function**. Any code inside its curly brackets {} will be executed.

Line 5: cout (pronounced "see-out") is an **object** used together with the *insertion operator* (<<) to output/print text. In our example it will output "Hello World".

Note: Every C++ statement ends with a semicolon ;.

Note: The body of int main() could also been written as:
int main () { cout << "Hello World! "; return 0; }

Example explained…

Remember: The compiler ignores white spaces. However, multiple lines makes the code more readable.

Line 6: return 0 ends the main function.

Line 7: Do not forget to add the closing curly bracket } to actually end the main function.

C++ Output

The cout object, together with the << operator, is used to output values/print text:

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

C++ Output…

You can add as many cout objects as you want. However, note that it does not insert a new line at the end of the output:

```
#include <iostream>
using namespace std;
int main() {
  cout << "Hello World!";
  cout << "I am learning C++";
  return 0;
}
```

C++ New Lines

To insert a new line, you can use the \n character:

```
#include <iostream>
using namespace std;

int main() {
  cout << "Hello World! \n";
  cout << "I am learning C++";
  return 0;
}
```

C++ New Lines…

Tip: Two \n \n characters after each other will create a blank line:

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

C++ New Lines…

Another way to insert a new line, is with the endl manipulator:

```
#include <iostream>
using namespace std;

int main() {
  cout << "Hello World!" << endl;
  cout << "I am learning C++";
  return 0;
}
```

References

- https://beginnersbook.com/2017/08/cpp-data-types/
- https://www.geeksforgeeks.org/c-data-types/
- http://www.cplusplus.com/doc/tutorial/basic_io/
- https://www.geeksforgeeks.org/basic-input-output-c/
- https://www.w3schools.com/cpp/default.asp
- https://www.javatpoint.com/cpp-tutorial

THANK YOU