Lecture # 02

DEPARTMENT OF COMPUTER SCIENCE

C++ Programming

Computer Instructor: Muhammad Abdullah Orakzai

C++
Programming

Contents

2

C++ Comments

A well-documented program is a good practice as a programmer. It makes a program more readable and error finding become easier. One important part of good documentation is Comments.

- In computer programming, a comment is a programmer-readable explanation or annotation in the source code of a computer program

- Comments are statements that are not executed by the compiler and interpreter.

Types of comments

1. Single line comment
2. Multi-line comment

1) Single line Comment

- ❖ Represented as // double forward slash.

- ❖ It is used to denote single line comment.

- ❖ It apply comment to a single line only.

- ❖ It is refereed as C++-style comments as it is originally part of C++ programming.

1) Single line Comment…

Example

#include<iostream>

int main()   **Output:** Welcome to PFlab

{

   // Single line Welcome user comment

   cout<<"Welcome to PFlab";

   return 0;

}

2) Multi-line Comment

❖ Represented as /* any text */.

❖ Start with forward slash and asterisk (/*) and end with asterisk and forward slash (*/).

❖ It is used to denote multi-line comment. It can apply comment to more than a single line.

❖ It is referred as C-Style comment as it was introduced in C programming.

2) Multi-line Comment…

```
#include<iostream>

int main()

{

    /* Multi-line Welcome user comment

    written to demonstrate comments

    in C/C++ */

cout<<"Welcome to PF lab";

    return 0;

}
```

Output:
Welcome to PF lab

Single or multi-line comments?

It is up to you which you want to use. Normally, we use // for short comments, and /* */ for longer.

C++ Keywords

❖ The words that are used by the language for special purpose are called keywords.

❖ These are also called reserved words. For example in a C++ the word main is used to indicate the starting of program, include is used to add header files, int is used to declare integer type variable. All these words are keywords of C++.

❖ The reserved words cannot be used as variable names in a program.

C++ Keywords…

A list of 32 Keywords in C++ Language which are also available in C language are given below.

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

C++ Keywords…

A list of 30 Keywords in C++ Language which are not available in C language are given below.

asm	dynamic_cast	namespace	reinterpret_cast	bool
explicit	new	static_cast	false	catch
operator	template	friend	private	class
this	inline	public	throw	const_cast
delete	mutable	protected	true	try
typeid	typename	using	virtual	wchar_t

Tokens

A program statement consist of variable names, keywords, constants, punctuations marks, operators etc. In C++ these elements of a statement are called tokens. In the following program segment:

main()

{

        int a, b;

}

main, {, }, int , a, b and punctuation marks (,) and (;) are tokens of the program.

Variables

❖ A named memory location where data is stored is called variable.

❖ A quantity whose value may change during execution of the program is called variable. It is represented by a symbol or name.

❖**Variable** is name of *reserved area allocated in memory*. In other words, it is a *name of memory location*.

❖It is a combination of "vary + able" that means its value can be changed.

❖**int** data=10;   // Here data is variable

Rules for Naming/Writing Variables

1. The first character of variable must be an alphabetic character.

2. Underscore can be used as first character of variable name.

3. Blank space are not allowed in a variable name.

4. Special character such as arithmetic operators, #, ^, cannot be used in a variable name.

5. The maximum length of a variable name depends upon the compiler of C++.

6. A variable name declared for one data type cannot be used to declare another data type.

Rules for Naming/Writing Variables…

C++ is a case sensitive language. Thus variable name with the same spellings but different cases are treated as different variable names. For example variable "Pay"

And "pay" are two different variables.

Rules for Naming/Writing Variables…_

❖ C++ is a case sensitive language: number is different from Number or nUmber

Valid identifiers:

❖ Int abc, char aBc, int first_var, float first

Invalid identifiers:

❖ Int 3bc, int a*b, int #a, int void

Rules for Naming/Writing Variables…_

camelCasevariable names:

number, firstName, dateOfBirth

PascalCasevariable names

Number, FirstName, DateOfBirth

Rules for Naming/Writing Variables…

Rule of Thumb

Use camelCasefor variablenames

Use PascalCasefor advanced naminge.g. Classes, functions

Rules for Naming/Writing Variables…

- int firstNumber;

- char gender = 'm';

- float piValue= 3.14;

- int myAge, int countryName;

- int a, b, c;

C++ Data Types

Data types specify the different sizes and values that can be stored in the variable.

Data Types tell us:

i) Type of data.

ii) Memory reserved (amount of memory).

iii) Range of value.

C++ Data Types…

Keyword	Range		Bytes of Memory
	Low	High	
char	-128	127	1
short	-32,768	32768	2
int	-2,147,483,648	2,147,483,647	4
long	-2,147,483,648	2,147,483,647	4
float	$3.4*10^{-38}$	$3.4*10^{38}$	4
double	$1.7*10^{-308}$	$1.7*10^{308}$	8
long double	$3.4*10^{-4932}$	$3.4*10^{4932}$	10

sizeof() Data Type

sizeof() function is used to find size (bytes) of data type

e.g: sizeof(char)

sizeof() Data Type…

```cpp
#include<iostream>

using namespace std;

int main()

{

    cout << "Size of char : " << sizeof(char)  << " byte" << endl;

    cout << "Size of int : " << sizeof(int) << " bytes" << endl;
```

sizeof() Data Type…

cout << "Size of short int : " << sizeof(short int) << " bytes" <<endl;

cout << "Size of long int : " << sizeof(long int) << " bytes" << endl;

cout << "Size of signed long int : " << sizeof(signed long int)

    << " bytes" << endl;

cout << "Size of unsigned long int : " << sizeof(unsigned long int)

    << " bytes" << endl;

```
cout << "Size of float : " << sizeof(float)  << " bytes" <<endl;


cout << "Size of double : " << sizeof(double)  << " bytes" << endl;
    system("pause");
  return 0;
}
```

sizeof() Data Type…

Output:

Size of char : 1 byte

Size of int : 4 bytes

Size of short int : 2 bytes

Size of long int : 8 bytes

sizeof() Data Type…

Output:

Size of signed long int : 8 bytes

Size of unsigned long int : 8 bytes

Size of float : 4 bytes

Size of double : 8 bytes

Integer Overflow

```cpp
int main()
{
    int n = 1000;
    cout << "n = " << n <<endl;
    n = n * 1000;
    cout << "n = "<< n <<endl;
    n = n * 1000;
    cout << "n = "<< n <<endl;
    n = n * 1000;
    cout << "n = " << n <<endl;
}
```

```
n = 1000;

n = 1000000;

n = 1000000000

n = -727379968
```

C++ Constant

When you do not want others (or yourself) to override existing variable values, use the const keyword (this will declare the variable as "constant", which means **unchangeable and read-only**)

Example

const int myNum = 15; // myNum will always be 15
myNum = 10; // error: assignment of read-only variable 'myNum'

C++ Constant…

You should always declare the variable as constant when you have values that are unlikely to change:

C++ Constant…

```cpp
#include <iostream>

using namespace std;

int main() {

  const int minutesPerHour = 60;

  const float PI = 3.14;

  cout << minutesPerHour << "\n";

  cout << PI;

  return 0;

}
```

❖ You have already learned that cout is used to output (print) values. Now we will use cin to get user input.

❖ cin is a predefined variable that reads data from the keyboard with the extraction operator (>>).

❖ In the following example, the user can input a number, which is stored in the variable x. Then we print the value of x:

C++ Input…

```cpp
#include <iostream>

using namespace std;

int main() {

  int x;

  cout << "Type a number: "; // Type a number and press enter

  cin >> x; // Get user input from the keyboard

  cout << "Your number is: " << x;

  return 0;

}
```

Good To Know

❖ cout is pronounced "see-out". Used for **output**, and uses the insertion operator (<<)

❖ cin is pronounced "see-in". Used for **input**, and uses the extraction operator (>>)

Creating a Simple Calculator

```cpp
#include <iostream>

using namespace std;

int main() {

  int x, y;

  int sum;

  cout << "Type a number: ";

  cin >> x;
```

```
 cout << "Type another number: ";

  cin >> y;

  sum = x + y;

  cout << "Sum is: " << sum;

  return 0;

}
```

```
Type a number: 2
Type another number: 4
Sum is: 6
```

Using Character

```cpp
#include<iostream>
using namespace std;
int main()
{
    char x;
    x = 'a';
    cout << "The character is: << x << endl;
}
```

```
The character is:    a
```

Using Character…

```cpp
#include<iostream>
using namespace std;
int main()
{
    char x;
    cout << "Enter Character:";
    cin >> x ;
    cout << "You entered " << x << endl;
}
```

```
Enter Character:        a
You entered             a
```

Using Character…

```cpp
#include<iostream>
using namespace std;
int main()
{
    char x;
    cout << "Enter Character:";
    cin >> x ;
    cout << "ASCII Value:" << int(x);
}
```

```
Enter Character:          a
ASCII Value:              97
```

Tasks

1. Write a C++ Program to swap two numbers without using the third variable.

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