Introduction to C++

1st part, introduction to c++

By: Muqaddar Niazi

Introduction to C++

C++ is a general-purpose programming language. It was created by Bjarne Stroustrup at Bell Labs circa 1980. C++ is very similar to C (invented by Dennis Ritchie in the early 1970s). C++ is so much compatible with C that it will probably compile over 99% of C programs without changing a line of source code. Though, C++ is a lot well-structured and safer language than C as it OOPs based.

Some computer languages are written for a specific purpose. Like, Java was initially devised to control toasters and some other electronics. C was developed for programming OS. Pascal was conceptualized to teach proper programming techniques. But C++ is a general-purpose language. It well deserves the widely acknowledged nickname "Swiss Pocket Knife of Languages."

Who uses c++?

Some of today's most visible used systems have their critical parts written in C++.

Examples:

Amadeus (airline ticketing)

Bloomberg (financial formation),

Amazon (Web commerce), Google (Web search)

Facebook (social media)

Continue...

Many programming languages depend on C++'s performance and reliability in their implementation. Examples include:

Java Virtual Machines

JavaScript interpreters (e.g., Google's V8)

Browsers (e.g., Internet Explorer, Mozilla's Firefox, Apple's Safari, and Google's Chrome)

Application and Web frameworks (e.g., Microsoft's .NET Web services framework).

Applications that involve local and wide area networks, user interaction, numeric, graphics, and database access highly depend on C++ language.

History of C++

C++ programming language was developed in 1980 by Bjarne Stroustrup at bell laboratories of AT&T (American Telephone & Telegraph), located in U.S.A.

Bjarne Stroustrup is known as the founder of C++ language. It was develop for adding a feature of OOP (Object Oriented Programming) in C without significantly changing the C component.

C++ programming is "relative" (called a superset) of C, it means any valid C program is also a valid C++ program.

C vs C++

No.	C	C++
1)	C follows the **procedural style programming.**	C++ is multi-paradigm. It supports both **procedural and object oriented.**
2)	Data is less secured in C.	In C++, you can use modifiers for class members to make it inaccessible for outside users.
4)	C does not support function overloading.	C++ supports function overloading.
5)	In C, you can't use functions in structure.	In C++, you can use functions in structure.
7)	In C, **scanf() and printf()** are mainly used for input/output.	C++ mainly uses stream **cin and cout** to perform input and output operations.
8)	Operator overloading is not possible in C.	Operator overloading is possible in C++.
10)	C does not provide the feature of namespace.	C++ supports the feature of namespace.
11)	Exception handling is not easy in C. It has to perform using other functions.	C++ provides exception handling using Try and Catch block.

C++ is an object oriented language

- C++ is object oriented programming language. OOPs makes development and maintenance easier where as in Procedure-oriented programming language it is not easy to manage if code grows as project size grows.

C++ Program

```
#include <iostream>
using namespace std;
void main()
 {
        cout << "Welcome to C++ Programming.";
}
```

Difference parts of the program

- #include<iostream> includes the standard input output library functions. It provides cin and cout methods for reading from input and writing to output respectively.
- main() The main() function is the entry point of every program in C++ language.

Continue...

- using namespace std;

A C++ program can be divided into different namespaces. A namespace is a part of the program in which certain names are recognized; outside of the namespace they're unknown. The directive using namespace std; says that all the program statements that follow are within the std namespace. Various program components such as cout are declared within this namespace. If we didn't use the using directive, we would need to add the std name to many program elements. For example, in the FIRST program we'd need to say

std::cout << "Every age has a language of its own.";

To avoid adding std:: dozens of times in programs we use the using directive instead.

Continue...

- cout << "Welcome to C++ Programming." is used to print the data "Welcome to C++ Programming." on the console.

Basic input/output

- C++ I/O operation is using the stream concept. Stream is the sequence of bytes or flow of data. It makes the performance fast.
- If bytes flow from main memory to device like printer, display screen, or a network connection, etc, this is called as output operation.
- If bytes flow from device like printer, display screen, or a network connection, etc. to main memory, this is called as input operation.

<iostream>

It is used to define the cout, cin and cerr objects, which correspond to standard output stream, standard input stream and standard error stream, respectively.

Standard Output (cout)

By default, the standard output of a program is the screen, and the C++ stream object defined to access it is cout. cout is used in conjunction with the insertion operator, which is written as << (two "less than" signs).

```
cout << "Output sentence"; // prints Output sentence on screen
cout << 120;               // prints number 120 on screen
cout << x;                 // prints the content of x on screen
```

Continue...

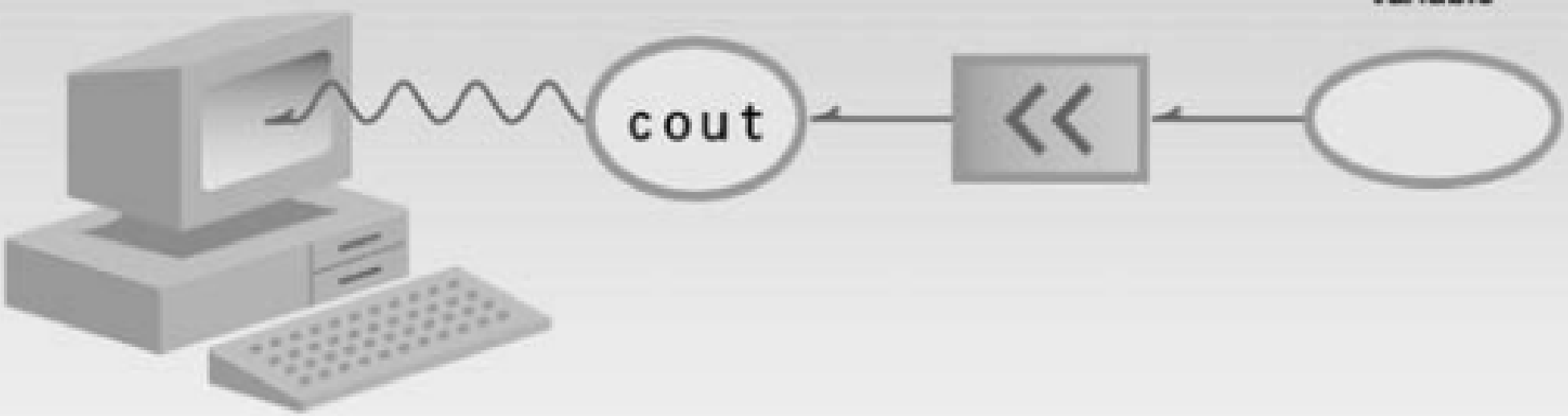
The identifier cout (pronounced “C out”) is actually an object. It is predefined in C++ to correspond to the standard output stream.

The operator << is called the insertion or put to operator. It directs the contents of the variable on its right to the object on its left. In FIRST it directs the string constant “Every age has a language of its own\n” to cout, which sends it to the display.

The insertion operator (<<) may be used more than once in a single statement:

```
cout << "Hello, " << "I am " << "a C++ statement";
```

Continue...



Continue...

For break line we can use “\n” and endl:

- In string format we can use “\n” Like:

```
cout<<“First line \n second line”;
```

- We can use the endl with new << operator for break line like:

```
cout<<“First line”<<endl<<“ Second line”<<endl;
```

Standard input (cin)

The keyword cin (pronounced “C in”) is an object, predefined in C++ to correspond to the standard input stream. This stream represents data coming from the keyboard

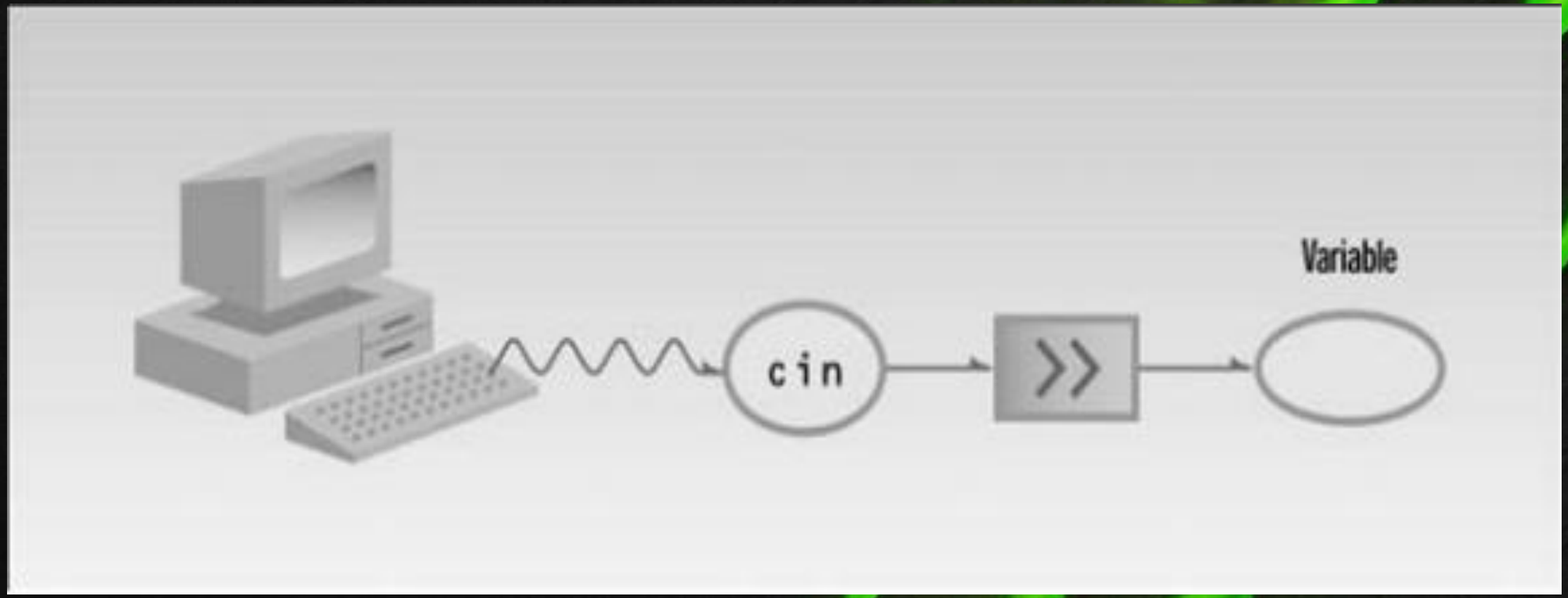
(unless it has been redirected). The >> is the extraction or get from operator. It takes the value

from the stream object on its left and places it in the variable on its right.

Example:

```
int age;
cin >> age;
```

Continue...



Basic input/output example

```
#include<iostream>
using namespace std;
main(){
        int age;
        cout<<“Enter Your age:”;
        cin>>age;
        cout<<“Your age is:”<<age<<endl;
}
```

Alert ⚠

As we have studied fundamental concepts of programming like: Arithmetic and logical operations, if else…, switch, loops ಠಠ, goto , array… in C language, these all concept are can be implemented in every programming language also in c++. With no change.

We will studied more concepts with cpp.

Next Topic

- Function

Good bye 👋.

End 🔚

- First part, introduction to c++.

References

www.javatpoint.com

Oop programming in c++, by: Robert lafore.