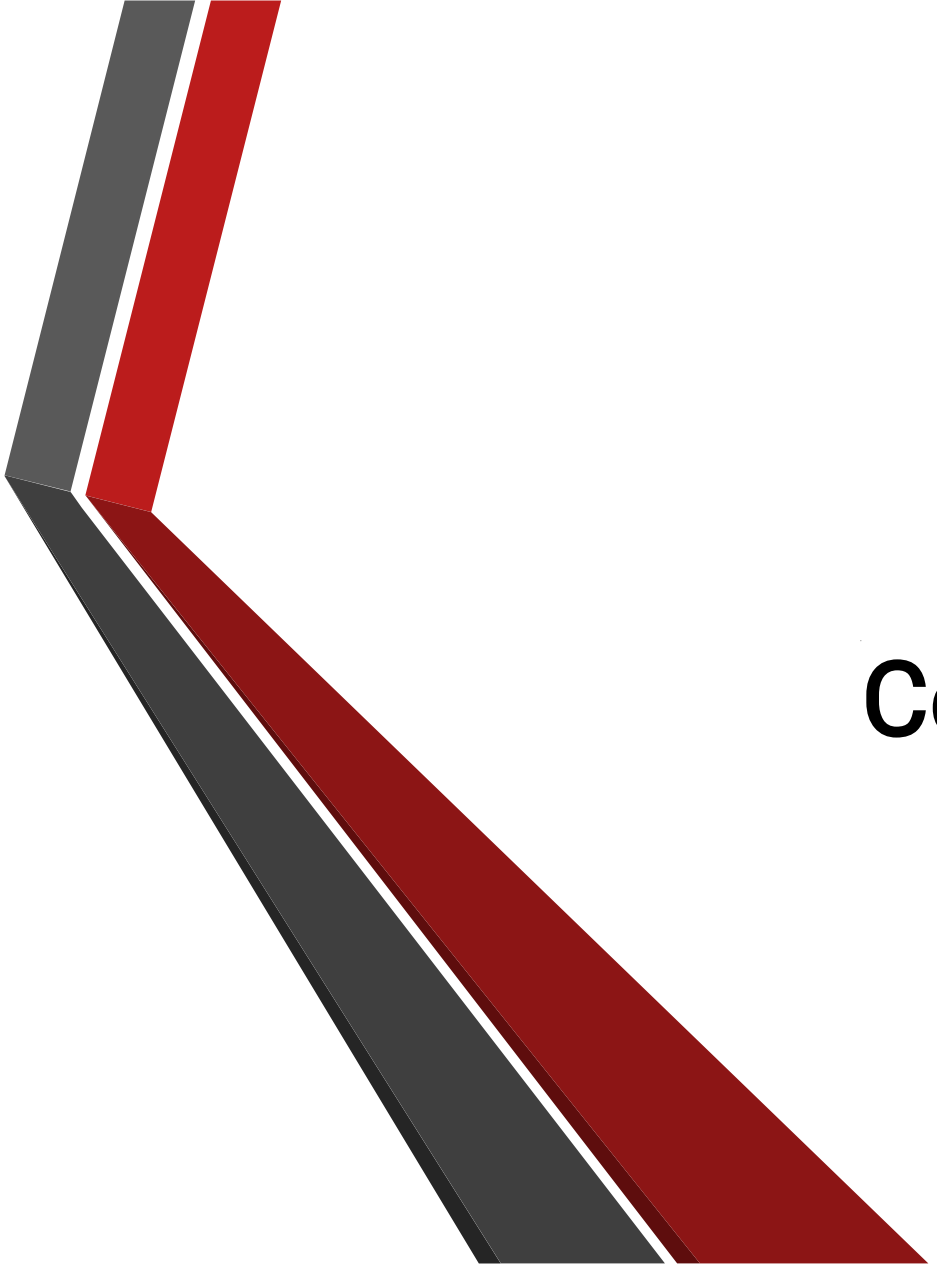


Computer Programming

Contents

- Anatomy of C++ basic program *(LL 04)*
- Outputting with *cout* statement *(LL 02)*
- *getch()* function *(LL 02)*
- *main* function *(LL 02)*
- *return* statement *(LL 02)*
- Header files *(LL 02)*
- Preprocessor directives *(LL 02)*
- Namespace *(LL 02)*

LL 02 = Learning Level 02 – Comprehension,

LL 04 = Learning Level 04 – Analysis

Anatomy of C++ basic program

- Every C++ program has an structure and is composed of fundamental components of the language.

- In order to understand the structure of a C++ program, consider the basic problem where we have to display the string **“14 Computer Systems”** on the monitor screen.

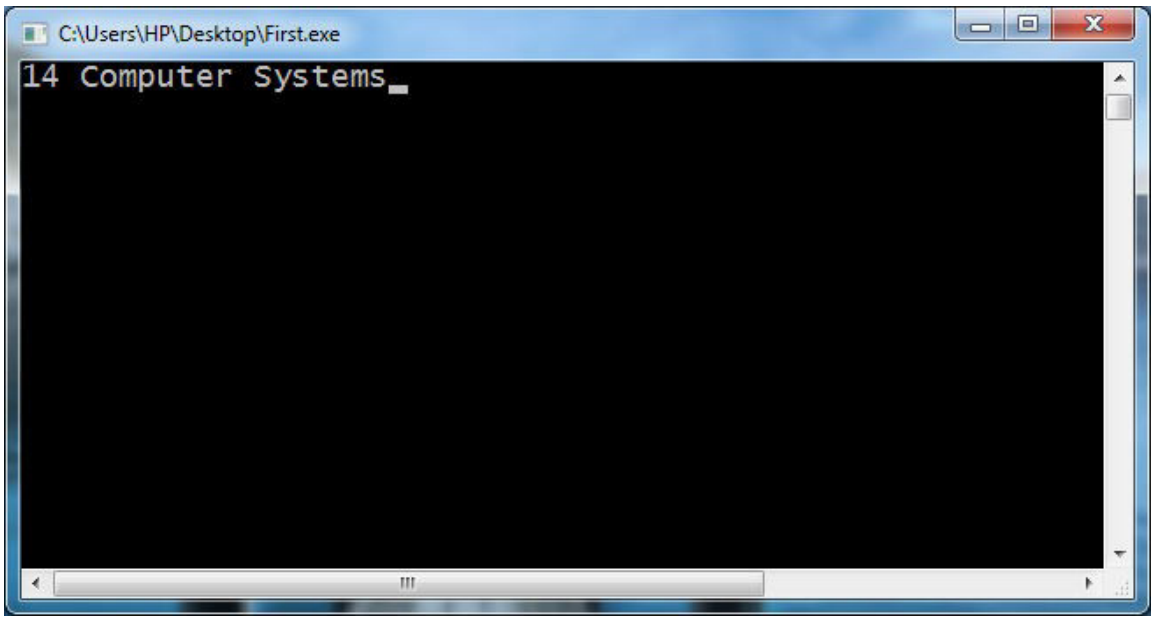
- Following C++ program does the said task.

Anatomy of C++ basic program

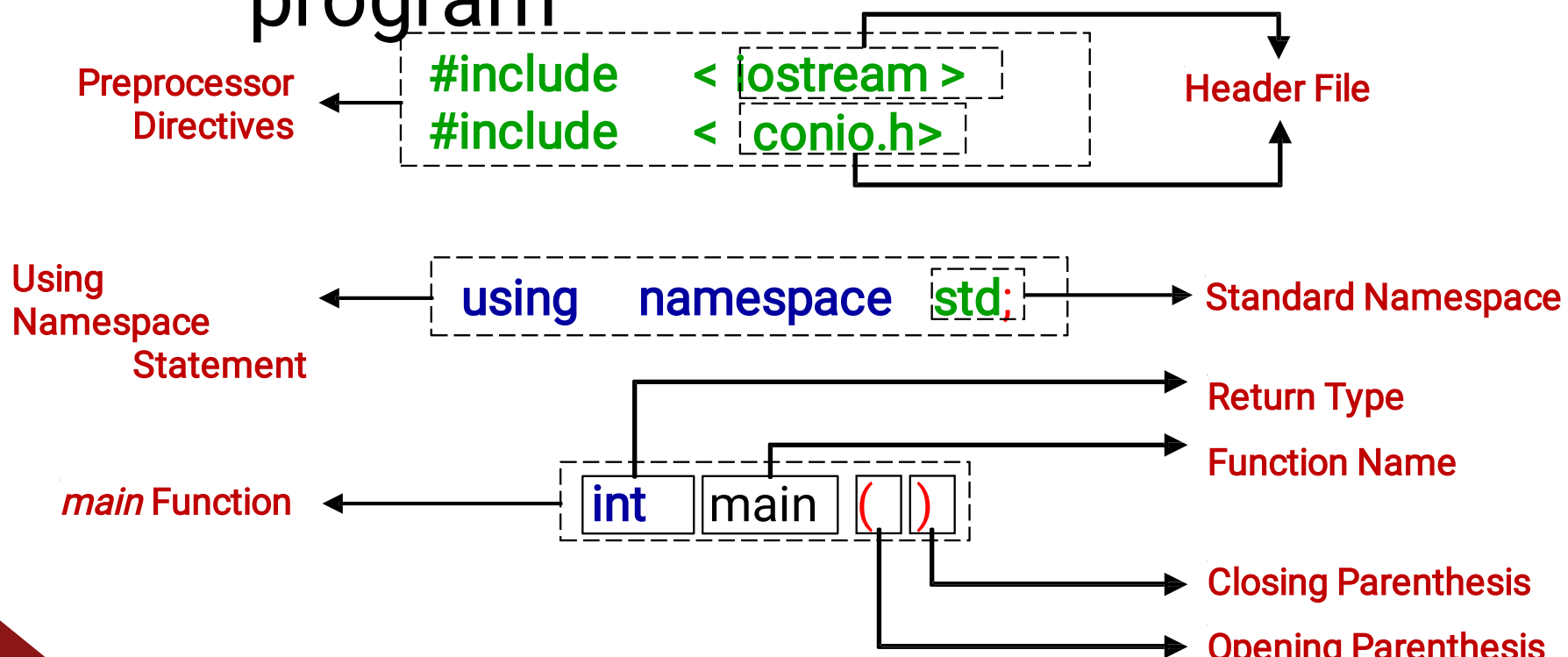
```
#include<iostream>
#include<conio.h>

using namespace std;

int  main()
{
cout<<"14  Computer  Systems"; getch();
return 0;
}
```

Anatomy of C++ basic program

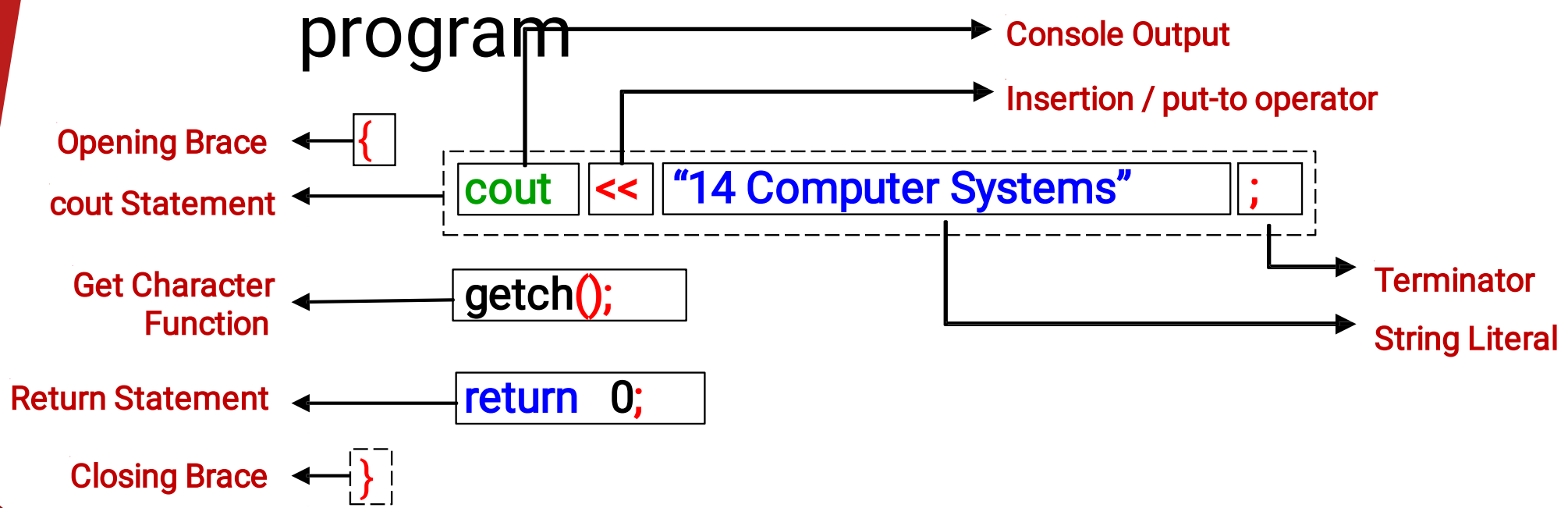
Output on monitor screen



Anatomy of C++ basic program



Anatomy of C++ basic program



Outputting with cout

cout << “14 Computer Systems”;

Outputting with cout

- In C++, cout statement is used to print/display the text or numbers on the monitor screen.
- Cout is the Standard Console Output, which is the monitor screen.
- Cout is used in conjunction with the **insertion operator** (**<<**).
- Any thing on the right side of **<<** will be displayed on monitor screen.
- **"14 Computer Systems"** is the string literal/constant to be displayed.
- **; (semicolon)** is called as **terminator**, it shows the end of any statement.

getch()function

- The getch function is used to get one character from the keyboard.
- Whenever the computer encounters getch function, it will wait for the user to press any key from the keyboard.
- Once the user presses any key on the keyboard that will be get by the getch function.
- In this program it is used to hold the screen so that we can view the output of the program.

getch()function

- The microprocessor is so fast that it can execute trillions of instructions in

one second.

- This simple program will be executed with in nano seconds.
- The computer will start the program, open up DOS black screen, print 14 Computer Systems string on it and will end after closing the black screen.
- This process will happen so quick (the screen will appear and quickly disappear with in nano seconds) that we even will not notice that the screen had actually appeared.

  In order to view the output we used getch function to hold the screen.

main()function

- Every C++ program must contain at least one function i.e. main function.
- Main function is the gateway of any C++ program because every program starts from the main function.
- The main is the name of the function.
- Every function name is followed by ( ) parenthesis.
- The int is the return type of the function, which specifies that, at the end, the main function will return one value to the operating system whose data type will be integer.
- The { and } specify the starting and ending of the function.

return Statement

- The return statement is used to return the control to another portion in the program or to the operating system.
- In this program, the return statement returns the value from the main function to the operating system.
- It can only return one value.
- It is mostly written as the last statement with in the function.
- When a Zero is returned to the operating system, it assumes that the program has executed Normally (Successfully).
- When any Non-Zero value is returned to the operating system, it assumes that the

program has executed Abnormally (Unsuccessfully).

- If you do not write the return statement in the main function, the compiler will automatically insert "return 0" in to the main function.

Header files

Input Output Stream

Console Input Output

Header file which are part of C as well as C++ *have* the extension .h Header files which are only the part of C++ *does not have* extension .h

Header files

- Header file is the file with extension **.h**
- It contains the declarations and definitions of the functions that can be shared between various different source programs.
- It is so called header file because it is always written at the top (head) of the program.
- When the compiler reads the **getch()** function, it does not know about how this function works.
- All the details (definition) of **getch()** function are stored in **conio.h** header file.
- If you want to use **getch()** function in your program you must include **conio.h** header file.

Preprocessor Directives

Pre ⇨ Before

Processor ⇨ Processing

Directives ⇨ Directions / Instructions / Commands

Instructions given before processing/compilation

Preprocessor Directives

- In C++, there is a special program called as **Preprocessor**.
- It executes all the instructions/commands before processing the actual program in the main function.
- All the instructions/commands/direction which are given to the preprocessor are known as **Preprocessor Directives**.
- In this basic program, the first two commands at the top are known as preprocessor directives.
- Preprocessor directives start with **#** (**hash**) sign.
- **#include** is known as **include directive**, which includes the header files inside the program before processing the actual program in the main function.

Namespaces

- A namespace can be considered as the collection of names.
- It organizes all the related names under the same namespace.
- There are many namespaces available in C++, one of them is **std** (***standard namespace***).
- The names like **cout**, **cin**, **endl** are all organized/included in **std** namespace.
- It is used to prevent name conflicts.
- It is used for categorizing the names.

Namespaces

The names like **cout**, **cin**, **endl** can be used in either of two ways.

- **Non-fully qualified name**
- **Fully qualified name**

Namespaces

- In non-fully qualified name, the names can be used simply as cout, cin and endl.
- But we have to write the using namespace statement at the top after preprocessor directives.

```
#include<iostream>
#include<conio.h>

using   namespace std;

int    main()
{
cout<<"14Computer                Systems"<<endl;
cout<<"MUET"; getch();
return  0;
}
```

Namespaces

- In **fully qualified name**, every name will be prepended with namespace and :: (double colon sign) as std::cout, std::cin, std::endl.
- Here we do not have to write the using namespace statement.

```
#include<iostream>
#include<conio.h>

int   main()
{
std::cout<<"14  Computer  Systems"<<std::endl; std::cout<<“MUET";
getch(); return  0;
}
```