Week 3

Session 3: Utilizing iostream and getline for C++ program

Element 2: Apply standard input/output of data streams and output formatting

ECT 124: Writing Programs using C++

Performance criteria (PC) for E2

PC1: Write code to perform input and output of data using the keywords cin, cout and getline.

PC2: Store data obtained using input/output streams.

PC3: Manipulate data obtained using input/output streams.

In this lesson!

Learning objectives:

By the end of this lesson, the student should be able to:

- ✓ Write programs to perform input and output of data.
- ✓ Store and process data using appropriate variables.

Class Activity 1

```
1
2
3   #include <iostream>
4
5   using namespace std;
6
7   int main()
8   {
9       cout << "Hello world!" << endl;
10      return 0;
11  }
12
```

Header file

2Using the standard namespace

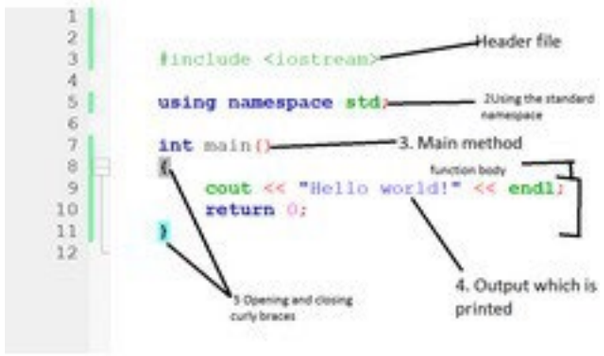
3. Main method

function body

4. Output which is printed

5 Opening and closing curly braces

Open Ended Question



What will be the printed output from the shown C++ program (previous slide)?

Please enter your answer here.

The input/output of data

- The C++ program has no built-in input or output (I/O) statements. Instead, I/O functions are provided by a library.

- Generally, the I/O statements are performed in the form of a sequence of bytes or known as streams. A stream is a sequence of bytes. We can think of it as an abstraction representing a device. You can perform I/O operations on the device via this abstraction.

- There are input stream and output stream.

Input stream

- If the direction of flow of bytes is from the device, for example from the keyboard to the computer memory then this process is called input.

Output stream

- If the direction of flow of bytes is opposite, which is for example from computer memory to device (display screen) then this process is called output.

The iostream header file

- To perform input and output operations in C++ program, we need to use the iostream header files. Without this header file, we cannot take input from the user or print any output.

- The iostream stands for standard input-output stream. It is a header file library that provides input and output functionality using streams.

- The command syntaxes of **cin** and **cout** are the most used standard input and output streams.

Standard Output Stream - cout

- It is an instance variable of the ostream class.

- It produces output on the standard output device such as the display screen. The syntax:

  cout << variable_name; OR **cout << "Any string literal";**

- We need to use the stream insertion operator << to insert data into the standard output stream cout for displaying on the screen.

- We can also use multiple stream insertion operators with a single **cout** to print multiple variables adjacent to each other on the same line.

```
cout << variable1 << variable2 << ... ;
```

- For printing the output on separate lines, we can write the syntaxes in 2 ways:

(1) Using **\n** (the new line character):

```
cout << variable1 << '\n' << variable2
```

(2) Using **endl** (a manipulator):

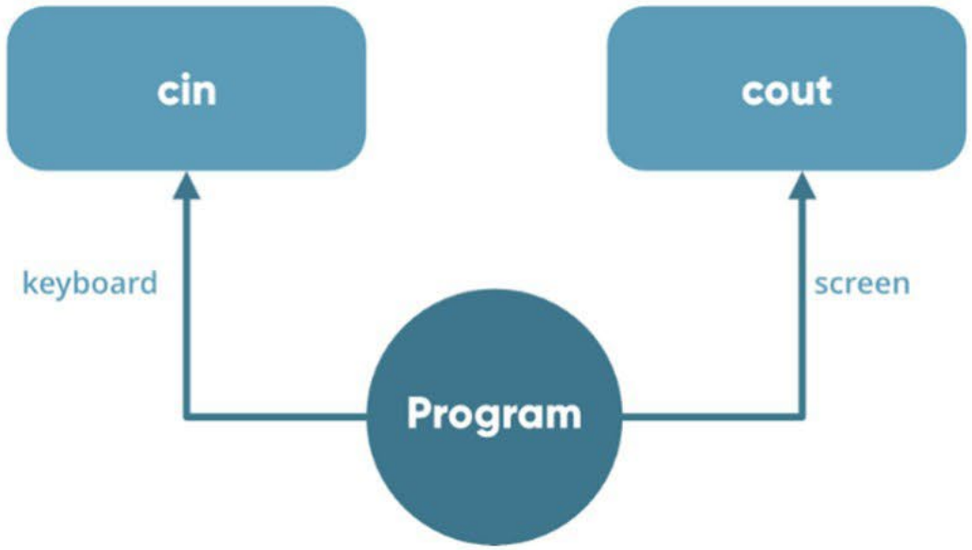
```
cout << variable1 << endl << variable2;
```

Standard Input Stream - cin

- It is an instance of the istream class.
- It reads input from the standard input device such as the keyboard. The syntax:

  cin >> variable_name; OR **cin >> variable1 >> variable2 >> ... ;**

- We need to use the stream extraction operator >> to extract data entered using the keyboard.
- The **cout** and **cin** of iostream class are the most basic methods of taking input and printing output in C++ program.
- To use **cout** and **cin** in C++ one must include the header file <iostream> in the program.



Example 1 Output stream using **cout**

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

int main()
{
    string name = "Ahmad Saeed";

    // You can use cout with a variable
    cout << name << endl;

    // Or you can also insert the output directly into the screen
    cout << "\nI am diploma student at HCT";

    return 0;
}
```

```
C:\Users\msharizal\OneDrive - Higher C
Ahmad Saeed

I am diploma student at HCT
--------------------------------
Process exited after 0.03992 sec
Press any key to continue . . .
```

Noticed the syntax usages of **endl** (to terminate the line) and **\n** to create a space between the two printed lines. Also the usage of **“”** to print anything in between the quotation mark.

Example 2 Output stream using **cout**

To print the numbers and character variables, we use the same **cout** but <u>without</u> using quotation marks.

```
1   #include <iostream>
2   using namespace std;
3
4   int main()
5   {
6       int num1 = 70;
7       double num2 = 256.783;
8       char ch = 'A';
9
10      cout << num1 << endl;     // print integer
11      cout << num2 << endl;     // print double
12      cout << "character: " << ch << endl;    // print char
13      return 0;
14  }
```

```
C:\Users\msharizal\Documents\HCT_Au
70
256.783
character: A

--------------------------------
Process exited after 0.04419 s
```

The **endl** manipulator is used to insert a new line. That is why each output is displayed in a new line. The **<<** operator can be used more than once if we want to print different variables, strings and so on in a single statement. For example (line 12 of above) where:

```
cout << "character: " << ch << endl;
```

Example 3

Input stream using **cin**

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

int main()
{
    int num;
    cout << "Enter an integer: ";
    cin >> num;    // Taking input
    cout << "The number is: " << num;
    return 0;
}
```

```
C:\Users\msharizal\Documents\HCT_Aug
Enter an integer: 25
The number is: 25
--------------------------------
Process exited after 5.618 seco
Press any key to continue . . .
```

We used **cin >> num;** to take input from the user. The input is stored in the variable num. We use the >> operator with cin to take input.

The getline command

- The C++ **getline()** is a standard library function that is used to read a string or a line from an input stream. It is a part of the **<string>** header filr.

- The **getline()** function extracts characters from the input stream and appends it to the string object until the delimiting character is encountered. This is an inbuilt function that accepts single and multiple character inputs.

- When working with user input in C++, the **cin** object allows us to get input information from the user. But when we try to log out the user's input that has multiple values, it only returns the first character.

- This happens because the C++ compiler assumes that any white space terminates the program when getting the input. That is, "My name is Ahmad" would only return "My" when compiled.

Example 4

```
C:\Users\msharizal\Documents\HCT_Aug 2
Tell us your name:
Ahmad Saeed Abdulla

Your name is: Ahmad
--------------------------------
Process exited after 10.22 seconds
Press any key to continue . . .
```

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

int main()
{
    string name;

    cout << "Tell us your name: \n";

    //This prompts the user to input a string, for example Ahmad Abdulla Saeed
    cin >> name;

    cout << "\nYour name is: " << name;
}
```

Example 5

```cpp
#include <iostream>
#include<string> // Add header file named <string>
using namespace std;

int main()
{
    string name;

    cout << "Tell us your name: \n";

    //This prompts the user to input a string, for example Ahmad Abdulla Saeed
    getline(cin, name);  // getline is used instead of cin

    cout << "\nYour name is: " << name;
}
```

```
C:\Users\msharizal\Documents\HCT_Aug 20
Tell us your name:
Ahmad Saeed Abdulla

Your name is: Ahmad Saeed Abdulla
--------------------------------
Process exited after 10.45 seconds
Press any key to continue . . .
```

Example 4

```
C:\Users\msharizal\Documents\HCT_Aug 2
Tell us your name:
Ahmad Saeed Abdulla

Your name is: Ahmad
--------------------------------
Process exited after 10.22 seconds
Press any key to continue . . .
```

Only Ahmad is print out although the user entered Ahmad Saeed Abdulla. Because of using **cin**, the compiler stop compiling after detecting whitespace after Ahmad.

Example 5

```
C:\Users\msharizal\Documents\HCT_Aug 20
Tell us your name:
Ahmad Saeed Abdulla

Your name is: Ahmad Saeed Abdulla
--------------------------------
Process exited after 10.45 seconds
Press any key to continue . . .
```

Complete print out when the user entered Ahmad Saeed Abdulla. The compiler detect all string when using **getline** under the **<string>** header file.

- The C++ **getline()** is an in-built function defined in the **<string.h>** header file that allows accepting and reading single and multiple line strings from the input stream.

- In C++ program, the **cin** object also allows input from the user, but not multi-word or multi-line input. This is where the **getline()** function comes in handy.

- The function continues accepting inputs and appending them to the string until it encounters a delimiting character.

- Thus, you can use it to keep adding inputs for longer strings. Some applications include:

  (1) Taking full name

  (2) Taking details such as address and bio

  (3) Asking for any long-form or multi-line input

Class Activity 2

(https://www.onlinegdb.com/online_c++_compiler)

Class Activity 2: Type the C++ program as shown below using the onlinegdb web-compiler. Then, print-screen the program output only.

- We can also take multiple inputs as shown in below.

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

int main()
{
    char a;
    int num1;
    double num2;

    cout << "Enter a character, an integer number, and a decimal numbers: \n"<<endl;
    cin >> a >> num1 >> num2;

    cout << "\nCharacter: " << a << endl;
    cout << "\nInteger number: " << num1 << endl;
    cout << "\nDecimal number: " << num2;

    return 0;
}
```

Collaborate Board

Summary

1 I/O using the iostream header file

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

int main()
{
    int num;
    cout << "Enter an integer: ";
    cin >> num;    // Taking input
    cout << "The number is: " << num;
    return 0;
}
```

```
C:\Users\msharizal\Documents\HCT_Aug
Enter an integer: 25
The number is: 25
--------------------------------
Process exited after 5.618 seco
Press any key to continue . . .
```

2 getline command

```cpp
#include <iostream>
#include<string> // Add header file named <string>
using namespace std;

int main()
{
    string name;

    cout << "Tell us your name: \n";

    //This prompts the user to input a string, for example Ahmad Abdulla Saeed
    getline(cin, name);  // getline is used instead of cin

    cout << "\nYour name is: " << name;
}
```

```
C:\Users\msharizal\Documents\HCT_Aug 20
Tell us your name:
Ahmad Saeed Abdulla

Your name is: Ahmad Saeed Abdulla
--------------------------------
Process exited after 10.45 seconds
Press any key to continue . . .
```

Higher Colleges of Technology

كليات التقنية العليا

Thank You

800 MyHCT (800 69428) www.hct.ac.ae HCT_UAE | hctuae