

Lecture -5 –

Programming Fundamentals Using C++

Your first C++ program

- A C++ program is a collection of commands or statements.
- Example:

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

int main()
{
    cout << "Hello world!";
    return 0;
}
```

Let's break down the parts of the code:

#include <iostream>

- C++ offers various headers, each of which contains information needed for programs to work properly.
- This program calls for the header <iostream>.
- The number sign (#) at the beginning of a line targets the compiler's pre-processor.
- In this case, #include tells the pre-processor to include the <iostream> header.
- The <iostream> header defines the standard stream objects that input and output data.
- The C++ compiler ignores blank lines.
- In general, blank lines serve to improve the code's readability and structure.
- Whitespace, such as spaces, tabs, and newlines, is also ignored, although it is used to enhance the program's visual attractiveness.

using namespace std;

- In our code, the line using namespace std; tells the compiler to use the std (standard) namespace.
- The std namespace includes features of the C++ Standard Library.

Main

- Program execution begins with the main function, int main().
- Curly brackets { } indicate the beginning and end of a function, which can also be called the function's body.
- The information inside the brackets indicates what the function does when executed.
- The entry point of every C++ program is main(), irrespective of what the program does.

cout

- The next line, cout << "Hello world!"; results in the display of "Hello world!" to the screen.
- In C++, streams are used to perform input and output operations.
- **cout** is used for the output operations.
- **cout** is used in combination with the insertion operator.
- Write the insertion operator as << to insert the data that comes after it into the stream that comes before.
- In C++, the semicolon is used to terminate a statement. Each statement must end with a semicolon. It indicates the end of one logical expression.

Statements

- A block is a set of logically connected statements, surrounded by opening and closing curly braces.
- For example:

```
{  
    cout << "Hello world!";  
    return 0;  
}
```

- You can have multiple statements on a single line, if you remember to end each statement with a semicolon.
- Failing to do so will result in an error.

Return

- The last instruction in the program is the return statement.
- The line `return 0;` terminates the `main()` function and causes it to return the value 0 to the calling process.
- A non-zero value (usually of 1) signals abnormal termination.

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

```
int main()  
{  
    cout << "Hello world!";  
    return 0;  
}
```

- If the return statement is left off, the C++ compiler implicitly inserts `"return 0;"` to the end of the `main()` function.
- **Note:** we can remove the `(return 0)` line from the code and still we get the same result.