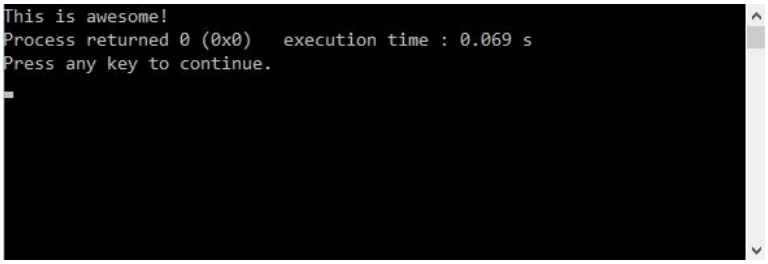
Lecture -6-

Your First C++ Program

- You can add multiple insertion operators after cout.

cout << "This " << "is " << "awesome!";



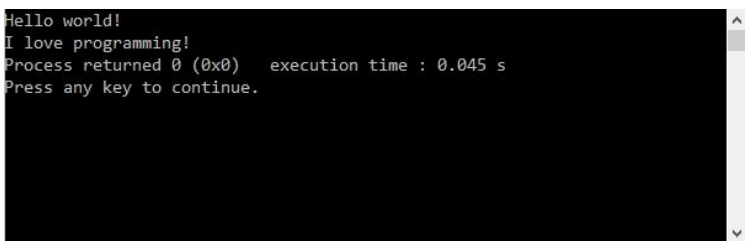
New Line

- The **cout** operator does not insert a line break at the end of the output.
- One way to print two lines is to use the **endl** manipulator, which will put in a line break.

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

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

- The **endl** manipulator moves down to a new line to print the second text.
- **Result:**



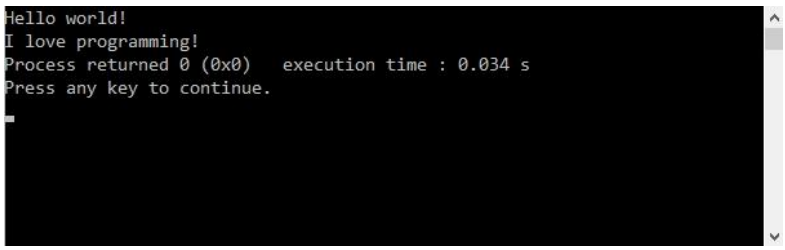
New Lines

- The new line character **\n** can be used as an alternative to **endl**.
- The backslash (\) is called an **escape character**, and indicates a "special" character.
- **Example:**

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

int main()
{
cout << "Hello world! \n";
cout << "I love programming!";
return 0;
}
```

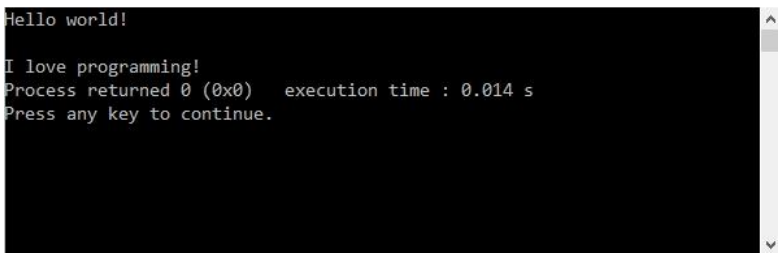
- **Result:**



Two newline characters placed together result in a blank line.

```
#include <iostream>
using namespace std;
int main()
{
cout << "Hello world! \n\n";
cout << "I love programming!";
return 0;
}
```

- **Result:**

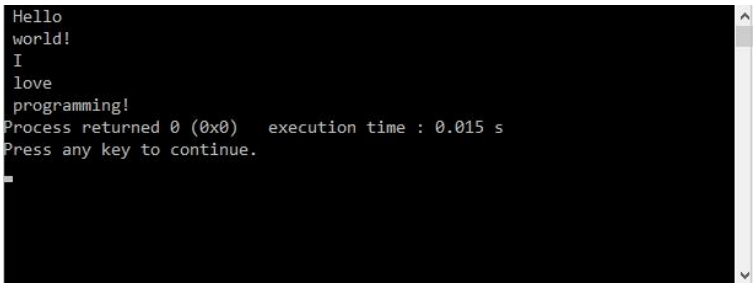


Multiple New Lines

- Using a single **cout** statement with as many instances of **\n** as your program requires will print out multiple lines of text.

```
#include <iostream>
using namespace std;
int main()
{
cout << " Hello \n world! \n I \n love \n
programming!";
return 0;
}
```

- **Result:**

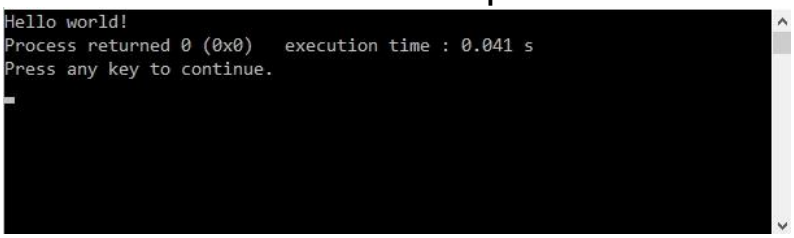


Comments

- **Comments** are explanatory statements that you can include in the C++ code to explain what the code is doing.
- The compiler ignores everything that appears in the comment, so none of that information shows in the result.
- A comment beginning with **two slashes (//)** is called a single-line comment.
- The slashes tell the compiler to ignore everything that follows, until the end of the line. **For example:**

```
#include <iostream>
using namespace std;
int main()
{
// prints "Hello world"
cout << "Hello world!";
return 0;
}
```

- When the above code is compiled, it will ignore the **// prints "Hello world"** statement and will produce the following result:



Multi-Line Comments

- Comments that require multiple lines begin with **/*** and end with ***/**
- You can place them on the same line or insert one or more lines between them.

```
/* This is a comment */

/* C++ comments can
span multiple lines
*/
```

Using Comments

- Comments can be written anywhere and can be repeated any number of times throughout the code.
- Within a comment marked with /* and */, // characters have no special meaning, and vice versa. This allows you to "nest" one comment type within the other.

```
/* Comment out printing of Hello world!

cout << "Hello world!"; // prints Hello world!

*/
```

- Adding comments to your code is a good practice.
- It facilitates a clear understanding of the code for you and for others who read it.