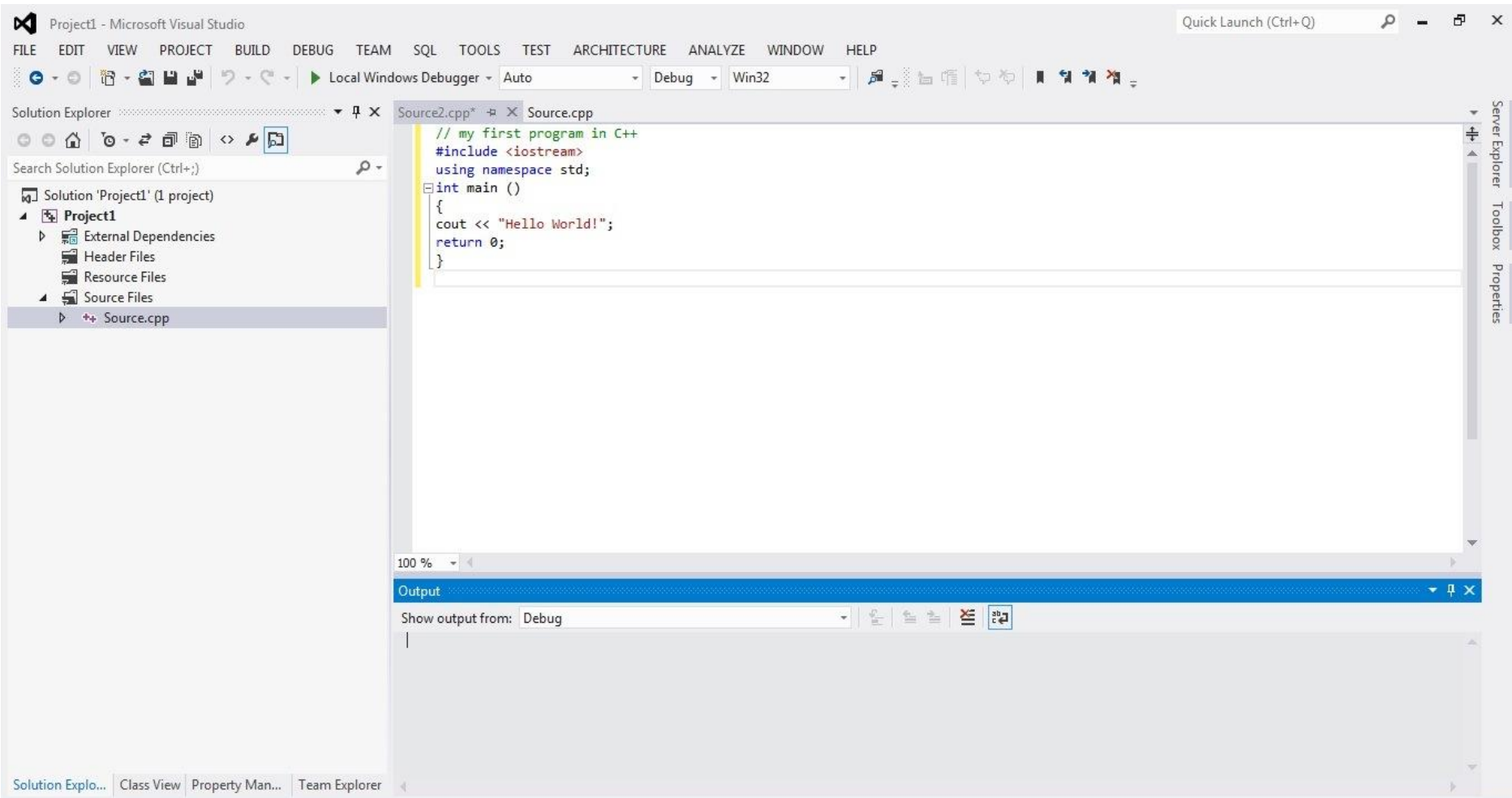
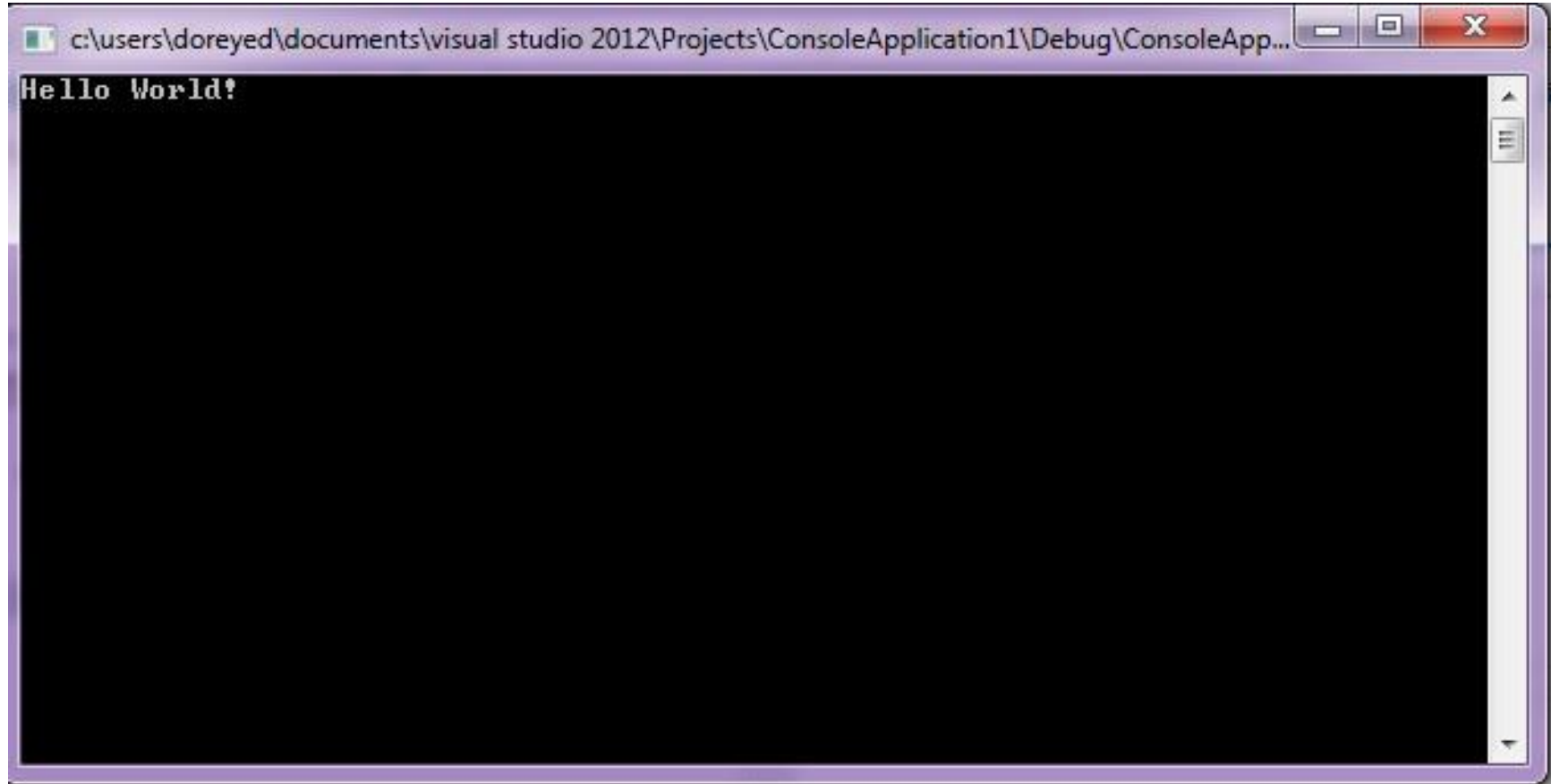


My first program in C++



```
// my first program in C++
#include <iostream>
using namespace std;
- int main ()
{
    cout << "Hello World!";
    return 0;
}
```

The output of My first program

A screenshot of a Windows console window. The title bar at the top shows the file path: c:\users\doreyed\documents\visual studio 2012\Projects\ConsoleApplication1\Debug\ConsoleApp... followed by standard window control buttons (minimize, maximize, close). The main area of the window is black, and the text "Hello World!" is printed in white at the top left. A vertical scrollbar is visible on the right side of the window.

```
c:\users\doreyed\documents\visual studio 2012\Projects\ConsoleApplication1\Debug\ConsoleApp...  
Hello World!
```

// my first program in C++

This is a comment line. All lines start with two slash signs (//) are considered comments and do not have any effect on the behavior of the program. The programmer can use them to include short explanations within the source code itself. In this case, the line is a brief description of what our program is.

include <iostream>

Lines beginning with a hash sign (#) are directives for the preprocessor. They are not regular code lines with expressions but indications for the compiler's preprocessor. In this case the directive **#include <iostream>** tells the preprocessor to include the **iostream** standard file. This specific file (**iostream**) includes the declarations of the basic standard input-output library in C++, and it is included because its functionality is going to be used later in the program.

using namespace std;

After you write this instruction, if the compiler sees string it will know that you may be referring to std::string, and if it sees vector, it will know that you may be referring to std::vector. (Provided that you have included in your compilation unit the header files where they are defined, of course.) If you *don't* write it, when the compiler sees string or vector it will not know what you are referring to. You will need to explicitly tell it std::string or std::vector, and if you don't, you will get a compile error.

int main ()

This line corresponds to the beginning of the definition of the main function. The main function is the point by where all C++ programs start their execution, independently of its location within the source code. the instructions contained within this function's definition will always be the first ones to be executed in any C++ program. For that same reason, it is essential that all C++ programs have a main function. The word main is followed in the code by a pair of parentheses (()). After these parentheses we can find the body of the main function enclosed in braces ({}). What is contained within these braces is what the function does when it is executed.

Parentheses	()
Square Brackets	[]
Angle Brackets	< >
Braces	{ }
Slashes	\ /

```
cout << "Hello World!";
```

This line is a C++ statement. A statement is a simple or compound expression that can actually produce some effect. In fact, this statement performs the only action that generates a visible effect in our first program. **cout** represents the standard output stream in C++, and the meaning of the entire statement is to insert a sequence of characters (in this case the **Hello World** sequence of characters) into the standard output stream. **cout** is declared in the **iostream** standard file within the **std namespace**, so that's why we needed to include that specific file and to declare that we were going to use this specific **namespace** earlier in our code.

Notice that the statement ends with a semicolon character (;). This character is used to mark the end of the statement and in fact it must be included at the end of all expression statements in all C++ programs (one of the most common syntax errors is indeed to forget to include some semicolon after a statement).

return 0;

The return statement causes the main function to finish. return may be followed by a return code (in our example is followed by the return code 0). A return code of 0 for the main function is generally interpreted as the program worked as expected without any errors during its execution. This is the most usual way to end a C++ console program.

You may have noticed that not all the lines of this program perform actions when the code is executed. There were lines containing only comments (those beginning by //). There were lines with directives for the compiler's preprocessor (those beginning by #). Then there were lines that began the declaration of a function (in this case, the main function) and, finally lines with statements (like the insertion into `cout`), which were all included within the block delimited by the braces ({}) of the main function.

The program has been structured in different lines in order to be more readable, but in C++, we do not have strict rules on how to separate instructions in different lines. For example, instead of

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

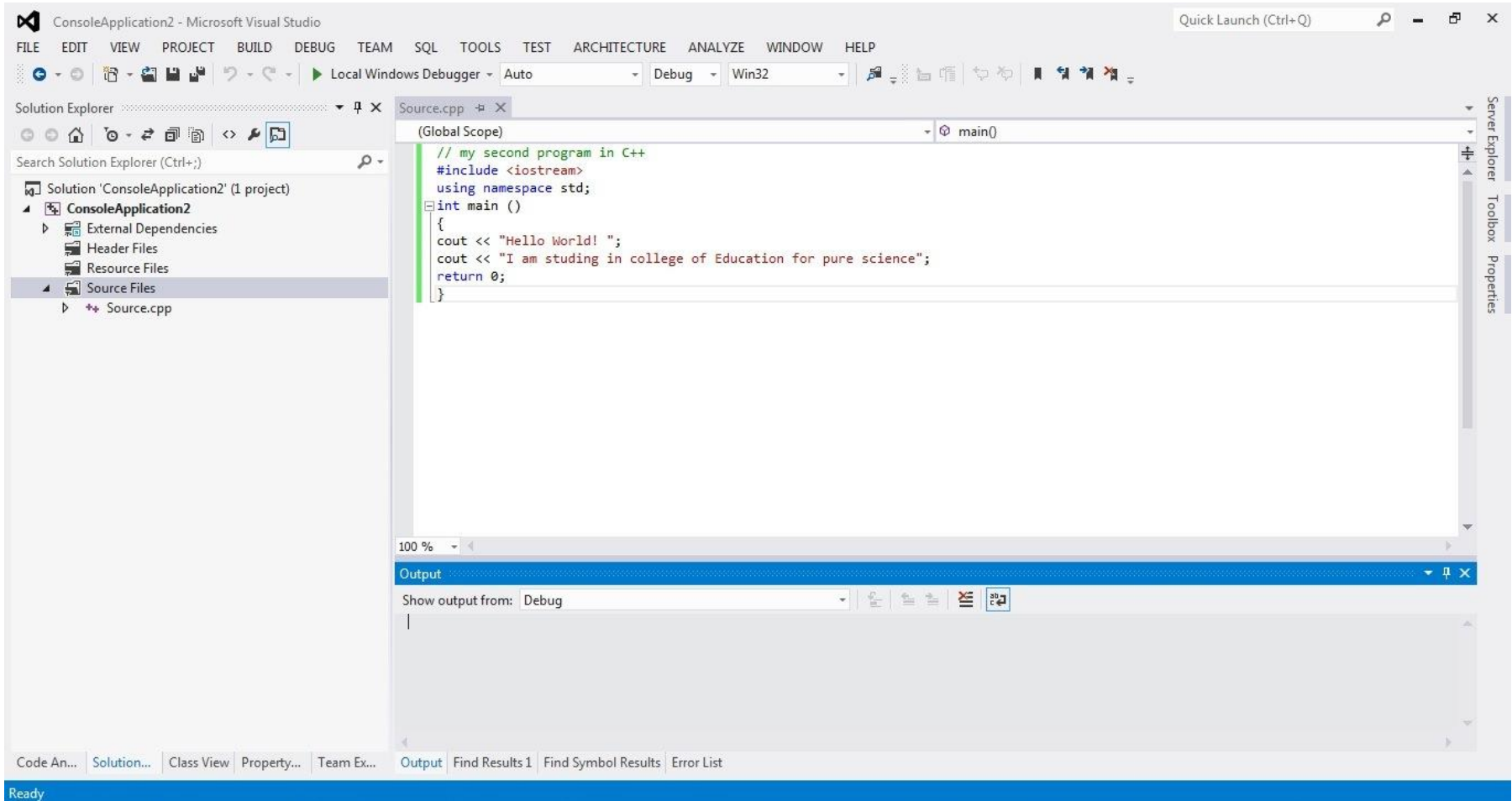
We could have written:

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

All in just one line and this would have had exactly the same meaning as the previous code.

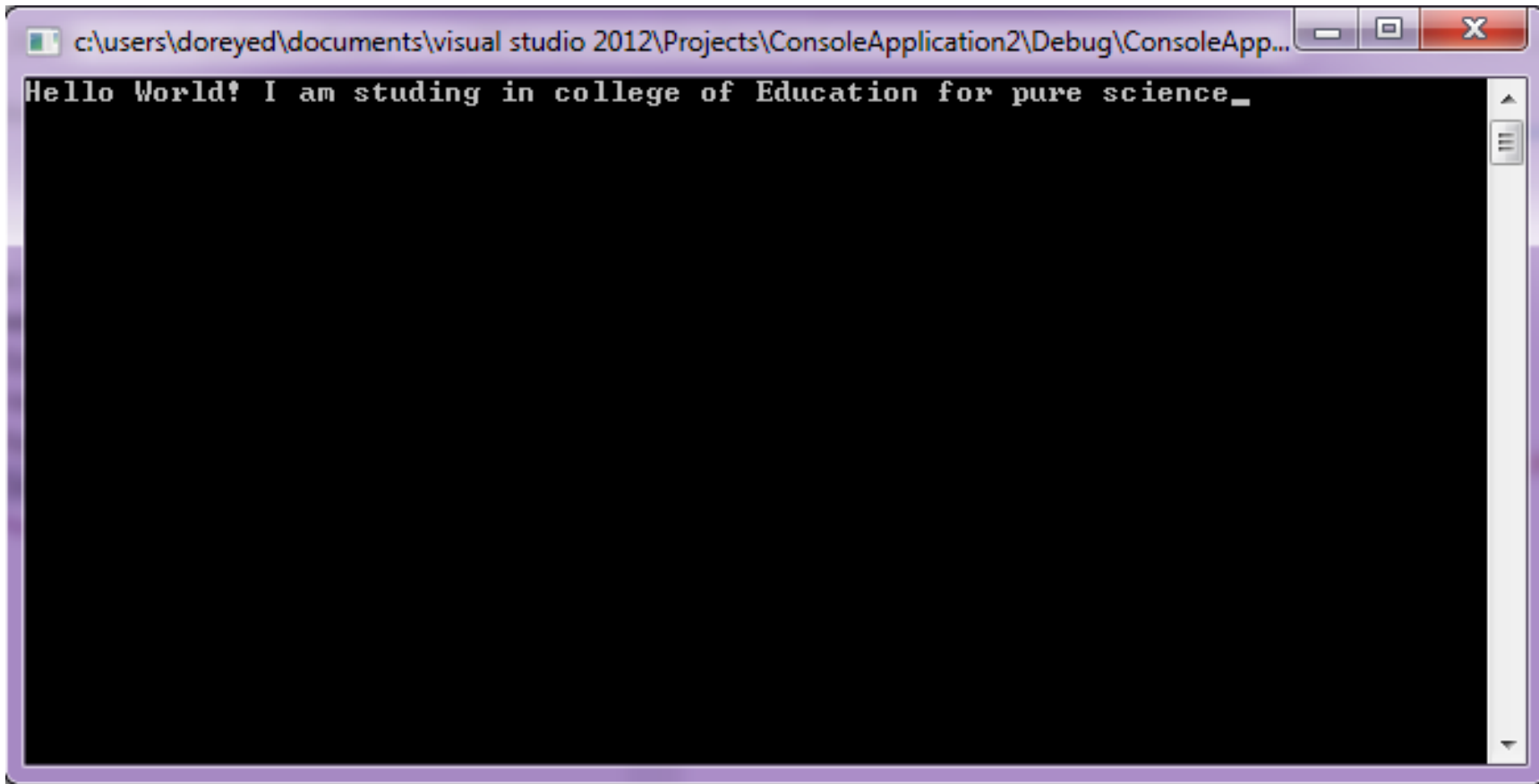
In C++, the separation between statements is specified with an ending semicolon (;) at the end of each one, so the separation in different code lines does not matter at all for this purpose. We can write many statements per line or write a single statement that takes many code lines. The division of code in different lines serves only to make it more readable and representation for the humans that may read it.

My Second program in C++



```
// my second program in C++
#include <iostream>
using namespace std;
int main ()
{
    cout << "Hello World! ";
    cout << "I am studing in college of Education for pure science";
    return 0;
}
```

The output of Second program



A screenshot of a Windows console window. The title bar shows the file path: `c:\users\doreyed\documents\visual studio 2012\Projects\ConsoleApplication2\Debug\ConsoleApp...`. The window contains a single line of text: `Hello World! I am studing in college of Education for pure science_`. The text is displayed in a monospaced font on a black background. The window has standard Windows controls (minimize, maximize, close) in the top right corner.

In this case, we performed two insertions into cout in two different statements. Once again, the separation in different lines of code has been done just to give greater readability to the program, since main could have been perfectly valid defined this way:

```
int main () { cout << "Hello World! "; cout << "I am studing in college of Education for pure science"; return 0; }
```

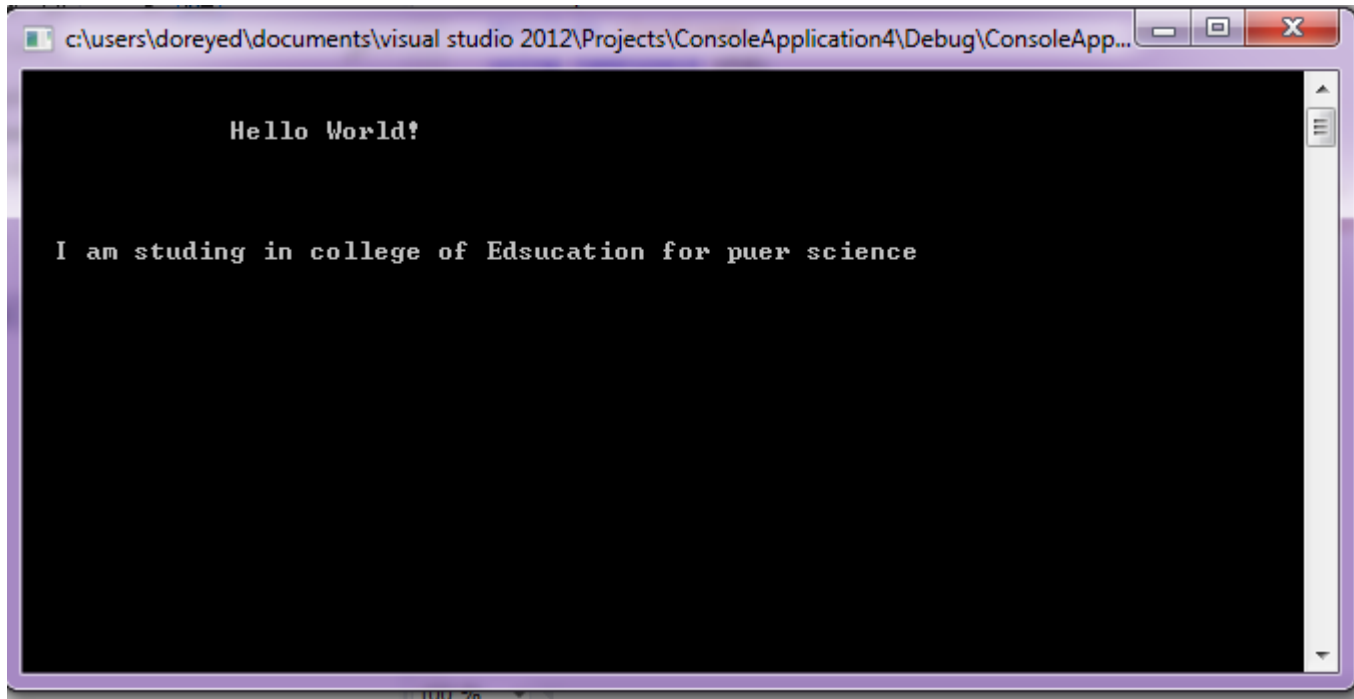
We were also free to divide the code into more lines if we considered it more convenient:

```
int main ()
{
    cout <<
        "Hello World! ";
    cout <<
        "I am studing in college of Education for pure science";
    return 0;
}
```

And the result would again have been exactly the same as in the previous examples.

New line \n

```
#include <iostream>
using namespace std;
int main ()
{
    cout << "          \n \n          Hello World! \n \n \n \n \n ";
    cout << " I am studing in college of Edsuction for puer science";
    return 0;
}
```



```
c:\users\doreyed\documents\visual studio 2012\Projects\ConsoleApplication4\Debug\ConsoleApp...
Hello World?

I am studing in college of Edsuction for puer science
```