CC-301 Programming Fundamentals

Lecture 6

Engr. Waseem Ullah Khan

Ternary Operator

Ternary operator (also known as the conditional operator) can be used to replace ***if...else***

A ternary operator evaluates the test condition and executes a block of code based on the result of the condition

Its syntax is

- ***condition ? expression1 : expression2;***

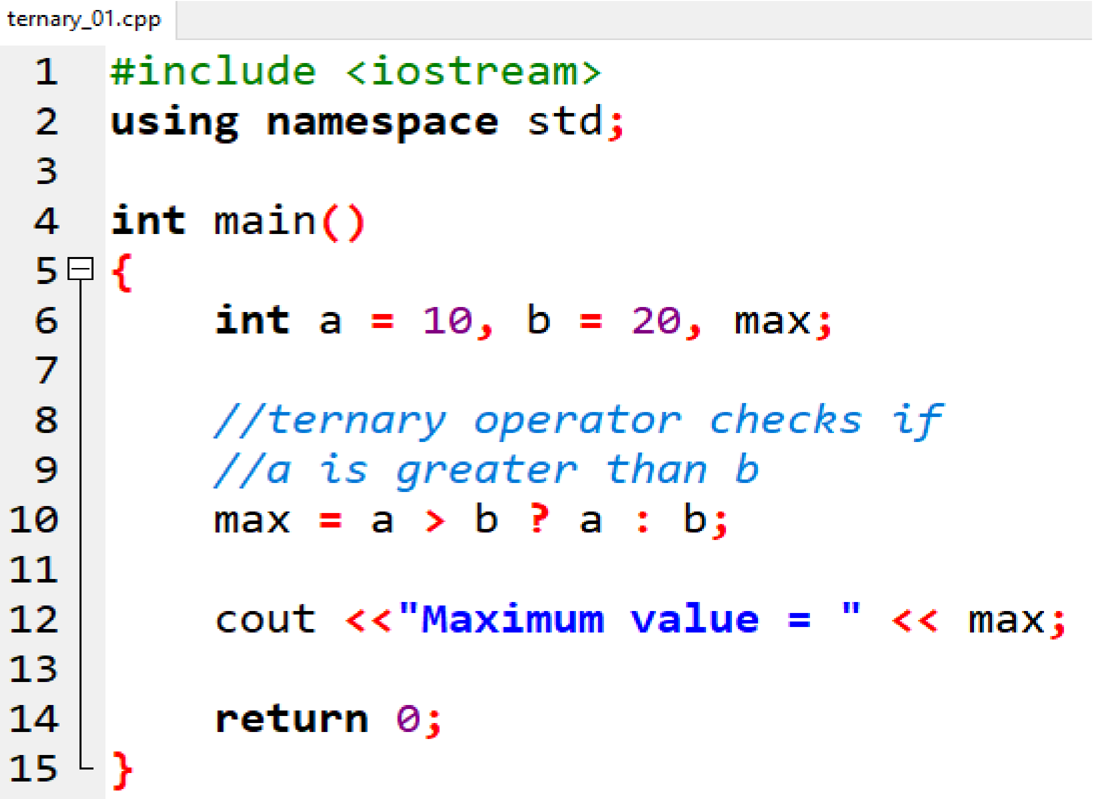
Here, condition is evaluated and

- **if condition is true,** expression1 is executed
- **if condition is false,** expression2 is executed

The ternary operator takes 3 operands ***(condition, expression1 and expression2)***. Hence, the name ternary operator

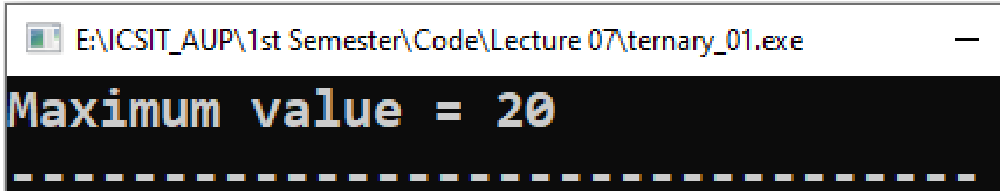
Example - Ternary Operator

Example: 01



ternary_01.cpp

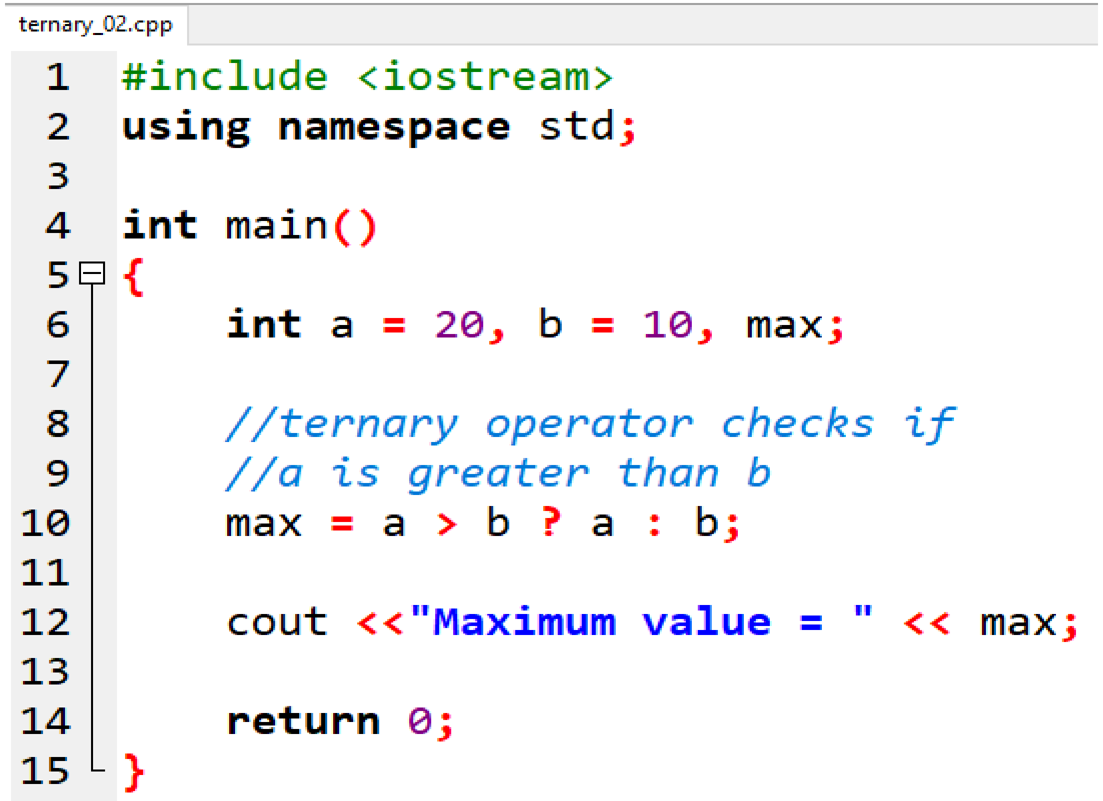
```cpp
#include <iostream>
using namespace std;

int main()
{
    int a = 10, b = 20, max;

    //ternary operator checks if
    //a is greater than b
    max = a > b ? a : b;

    cout <<"Maximum value = " << max;

    return 0;
}
```



E:\ICSIT_AUP\1st Semester\Code\Lecture 07\ternary_01.exe

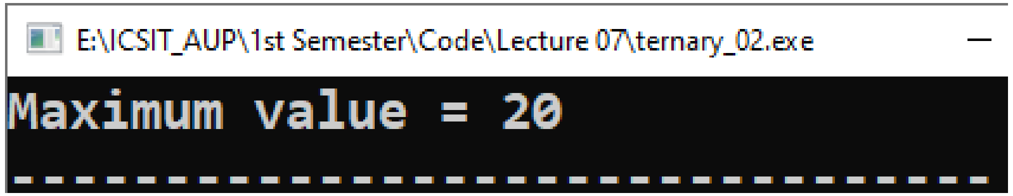
```
Maximum value = 20
--------------------------------
```

Example: 02



ternary_02.cpp

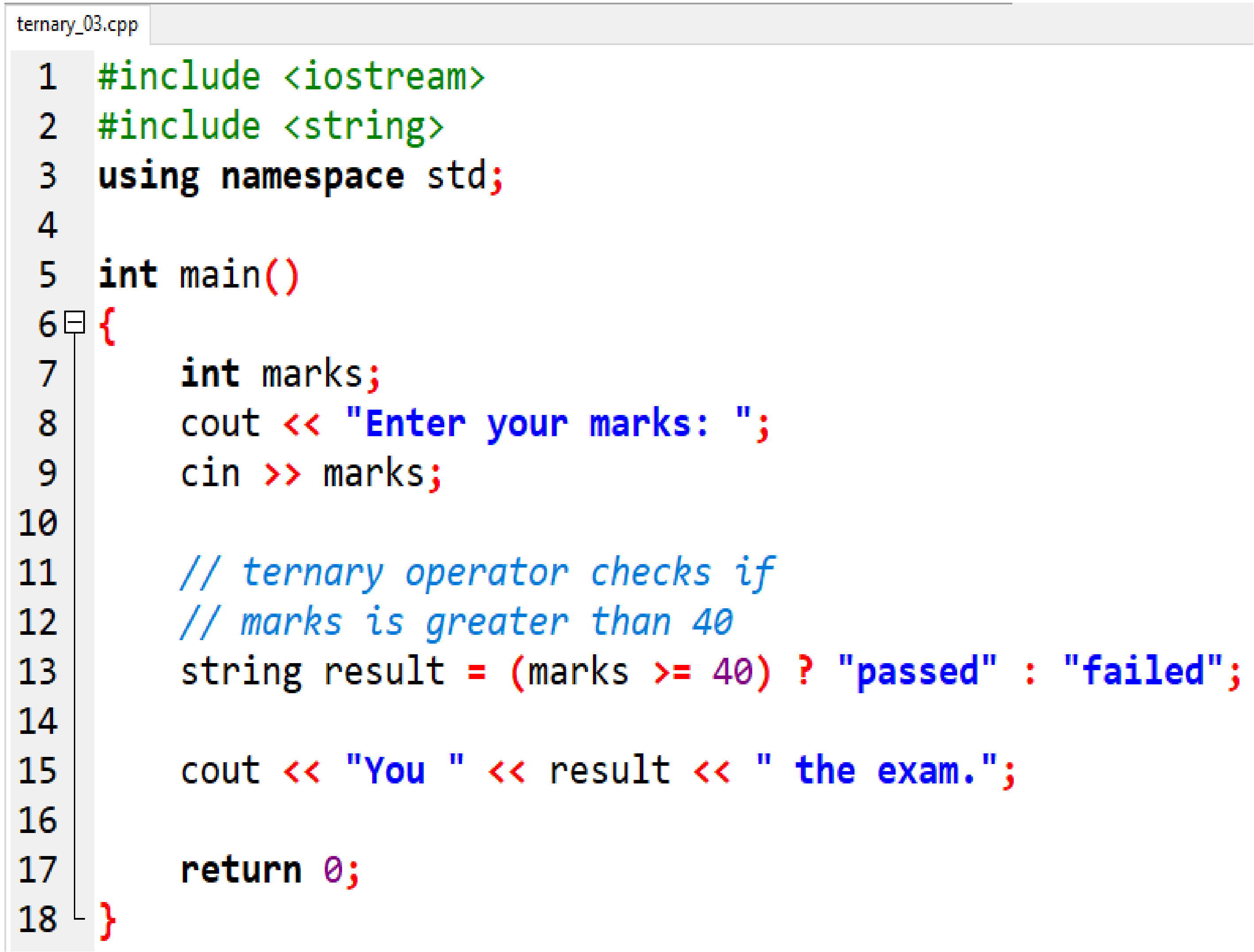
```cpp
#include <iostream>
using namespace std;

int main()
{
    int a = 20, b = 10, max;

    //ternary operator checks if
    //a is greater than b
    max = a > b ? a : b;

    cout <<"Maximum value = " << max;

    return 0;
}
```



E:\ICSIT_AUP\1st Semester\Code\Lecture 07\ternary_02.exe

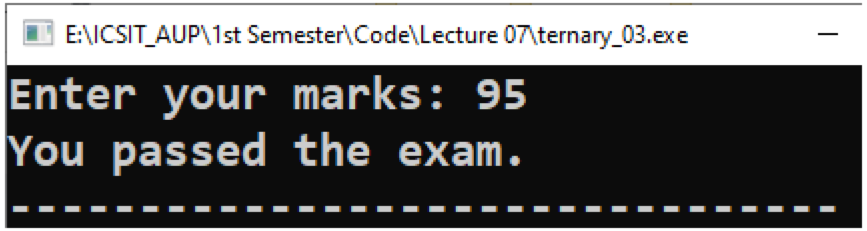
```
Maximum value = 20
--------------------------------
```

Example - Ternary Operator



ternary_03.cpp

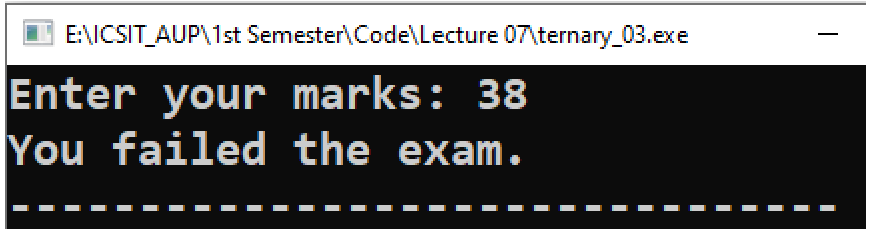
```cpp
#include <iostream>
#include <string>
using namespace std;

int main()
{
    int marks;
    cout << "Enter your marks: ";
    cin >> marks;

    // ternary operator checks if
    // marks is greater than 40
    string result = (marks >= 40) ? "passed" : "failed";

    cout << "You " << result << " the exam.";

    return 0;
}
```



E:\ICSIT_AUP\1st Semester\Code\Lecture 07\ternary_03.exe

```
Enter your marks: 95
You passed the exam.
--------------------------------
```

Suppose user enters 95. Then, condition marks >= 40 evaluates to true. Hence, first expression "passed" is assigned to result



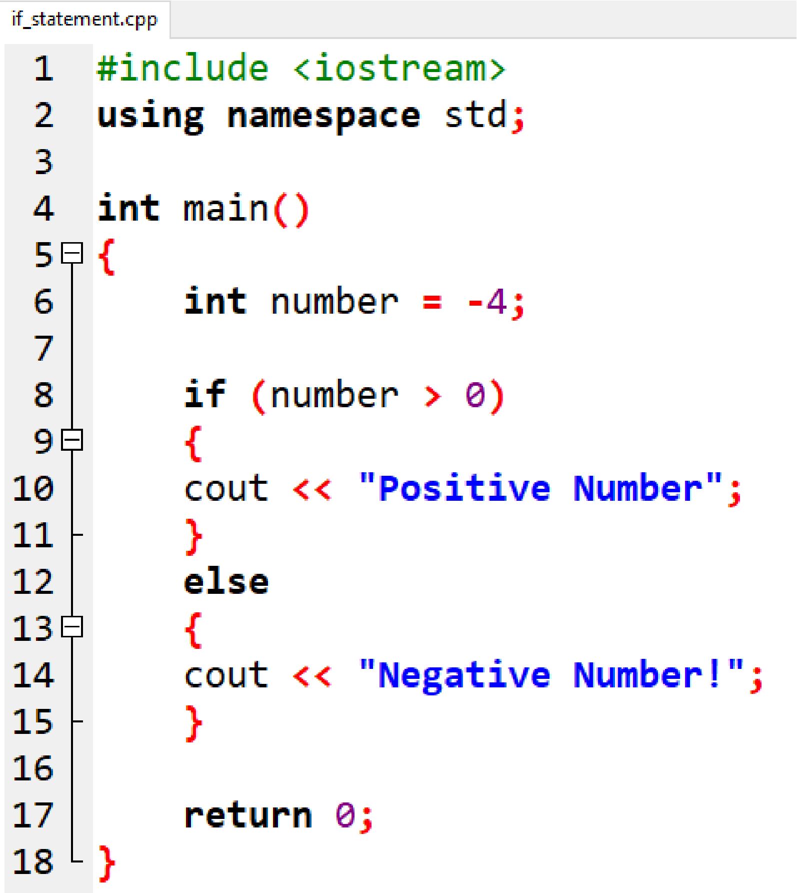
E:\ICSIT_AUP\1st Semester\Code\Lecture 07\ternary_03.exe

```
Enter your marks: 38
You failed the exam.
--------------------------------
```

Suppose user enters 38. Then, condition marks >= 40 evaluates to false. Hence, 2nd expression “failed" is assigned to result

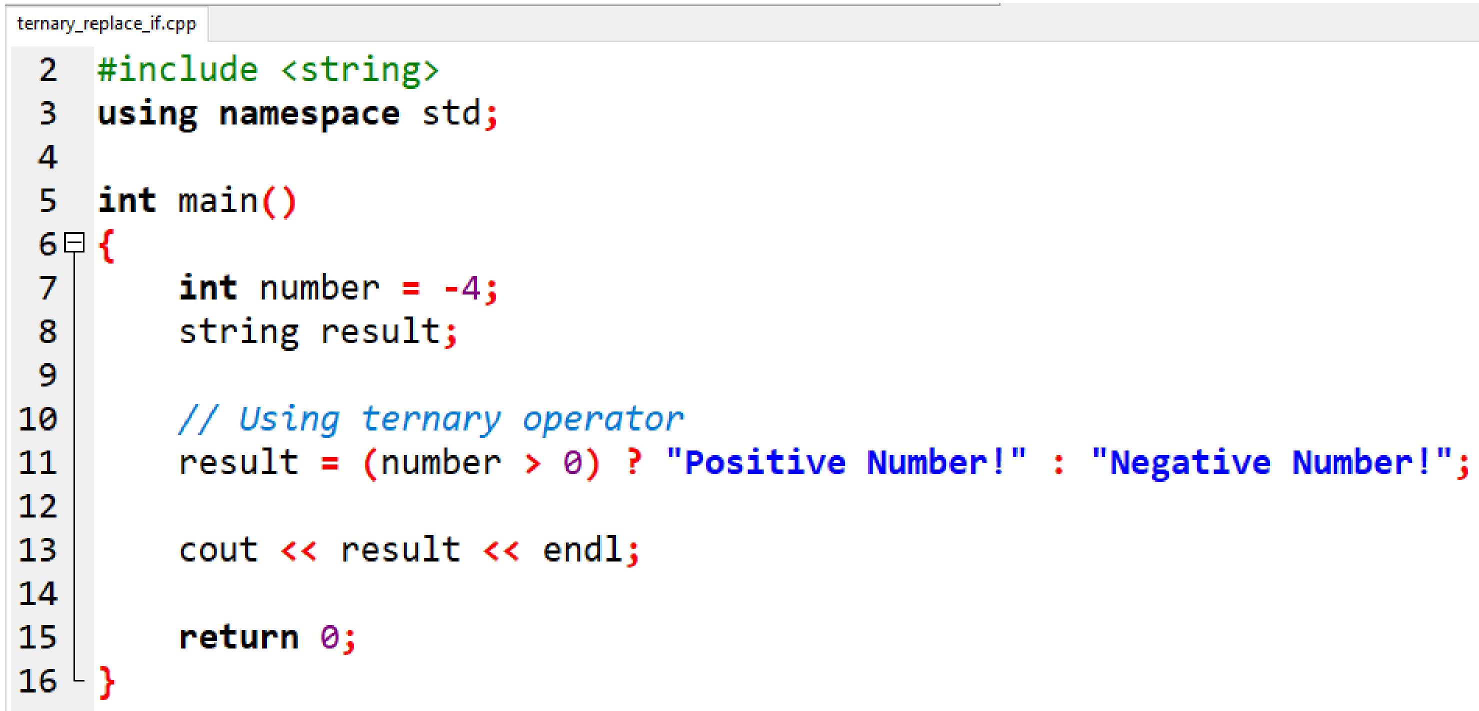
When to use a Ternary Operator?

In C++, the ternary operator can be used to replace certain types of ***if...else statements***



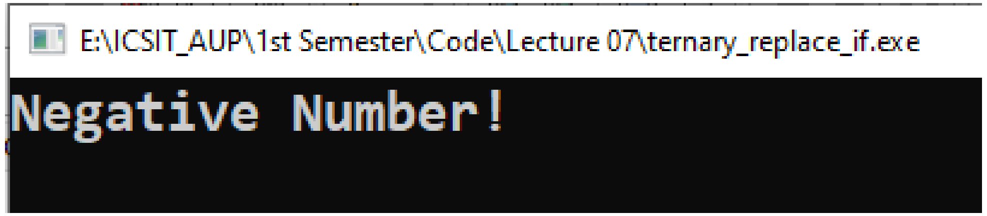
if_statement.cpp

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

int main()
{
    int number = -4;

    if (number > 0)
    {
    cout << "Positive Number";
    }
    else
    {
    cout << "Negative Number!";
    }

    return 0;
}
```



ternary_replace_if.cpp

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

int main()
{
    int number = -4;
    string result;

    // Using ternary operator
    result = (number > 0) ? "Positive Number!" : "Negative Number!";

    cout << result << endl;

    return 0;
}
```



E:\ICSIT_AUP\1st Semester\Code\Lecture 07\ternary_replace_if.exe

```
Negative Number!
```

Nested Ternary Operators

Ternary operators can be nested just like if-else statements

To use one ternary operator inside another ternary operator

Its syntax is

- **(condition1 ? expression1) : (condition2 ? expression2 : expression3);**

Here, condition is evaluated and

- if condition1 is true, expression1 is executed
- And, if condition1 is false, check condition2
- if condition2 is true, expression2 is executed
- if condition2 is false, expression3 is executed

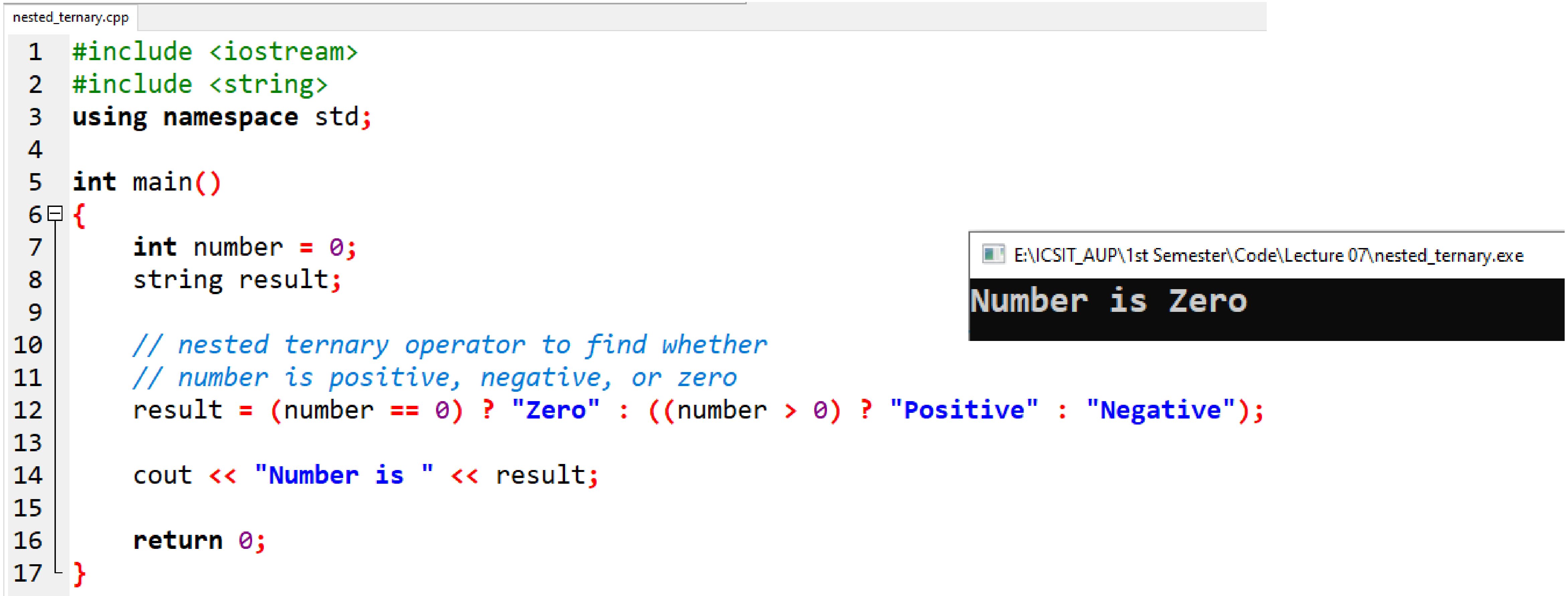
Using the nested if else statement to understand it better

```
if  (condition1)
        expression1
else if (condition2)
        expression2
else
        expression3
```

Nested Ternary Operators

Example:

Write C++ program to find whether number is positive, negative, or zero using nested ternary operator



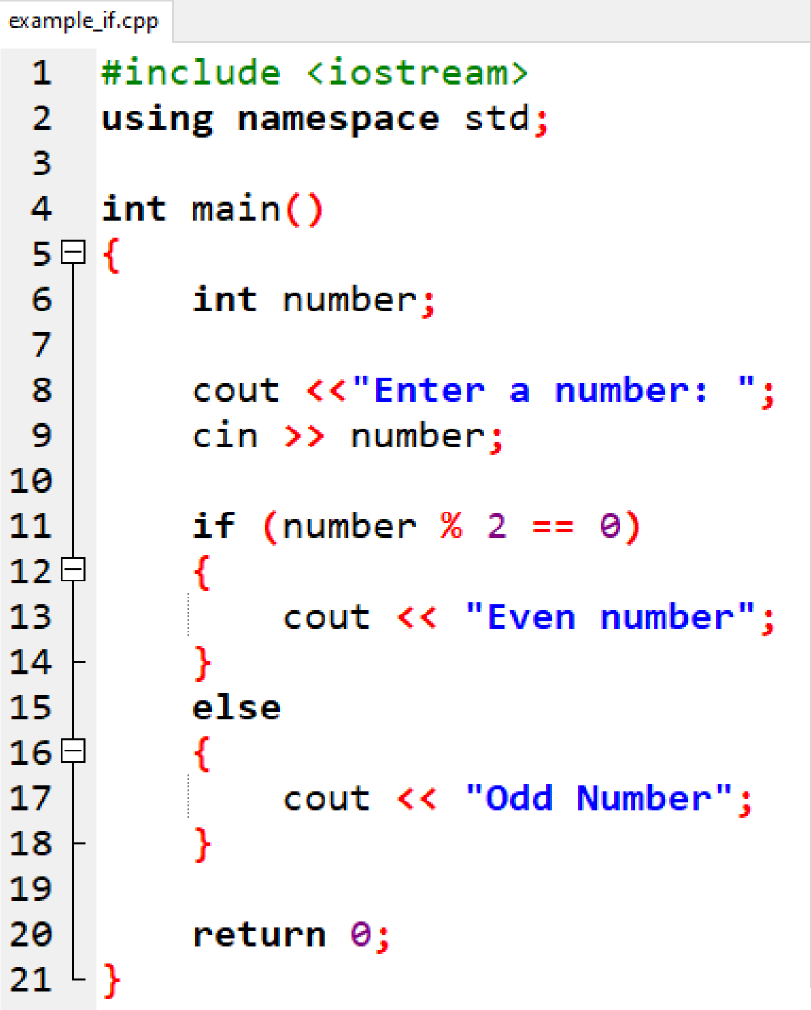
nested_ternary.cpp
E:\ICSIT_AUP\1st Semester\Code\Lecture 07\nested_ternary.exe

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

int main()
{
    int number = 0;
    string result;

    // nested ternary operator to find whether
    // number is positive, negative, or zero
    result = (number == 0) ? "Zero" : ((number > 0) ? "Positive" : "Negative");

    cout << "Number is " << result;

    return 0;
}
```

```
Number is Zero
```

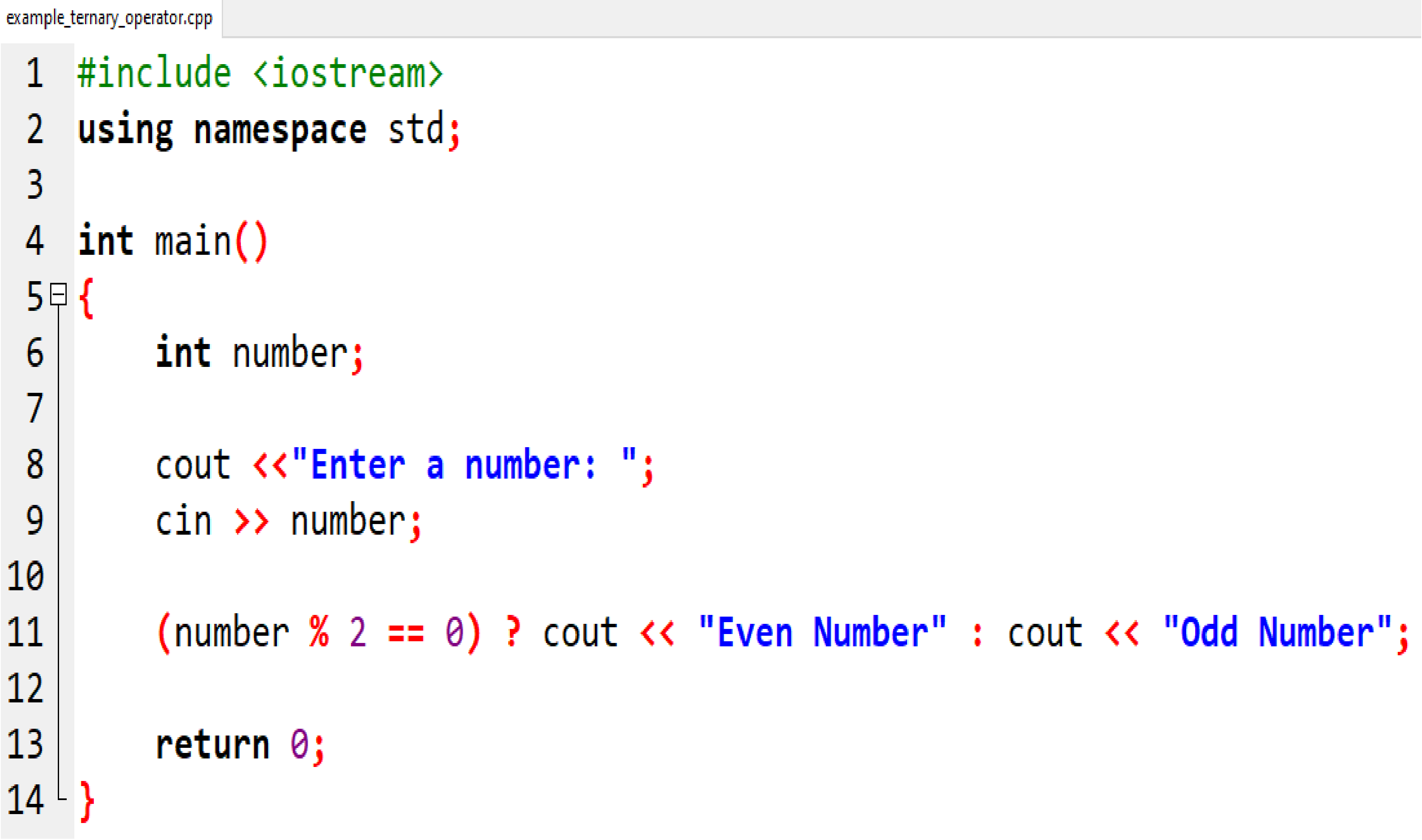
Exercise

C++ program to find even or odd number using ternary operator



example_if.cpp

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

int main()
{
    int number;

    cout <<"Enter a number: ";
    cin >> number;

    if (number % 2 == 0)
    {
        cout << "Even number";
    }
    else
    {
        cout << "Odd Number";
    }

    return 0;
}
```



example_ternary_operator.cpp

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

int main()
{
    int number;

    cout <<"Enter a number: ";
    cin >> number;

    (number % 2 == 0) ? cout << "Even Number" : cout << "Odd Number";

    return 0;
}
```