كليــات التقنيــة العليــا
Higher Colleges of Technology

The Campus of Tomorrow

Week 4

Session 4: Control structures using if statements

Element 3: Apply control structures using ‘if’ and ‘switch’ statements

ECT 124: Writing Programs using C++

Performance criteria (PC) for E3

PC1: Write applications using 'if' control structures.

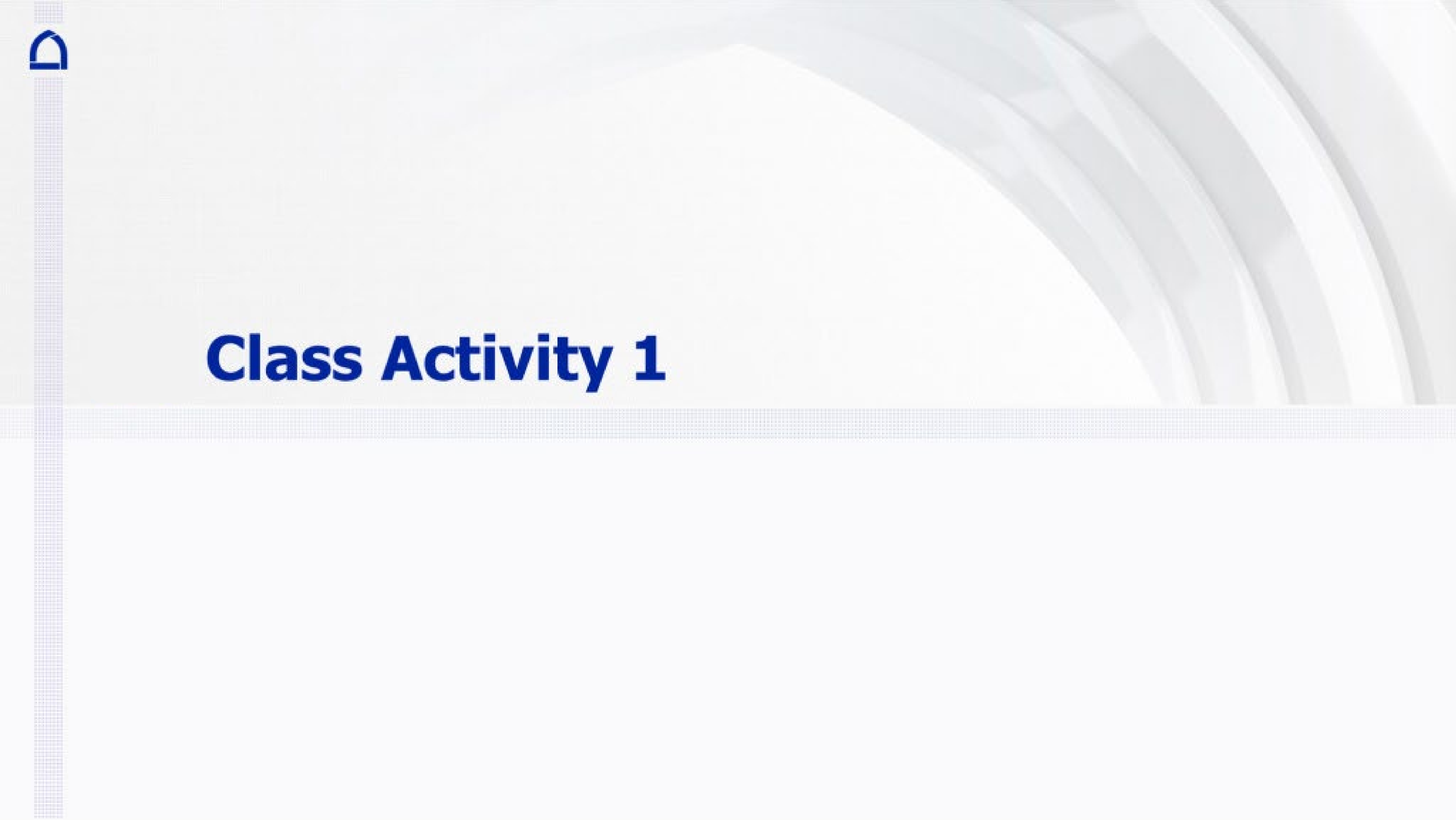
In this lesson!

PC2: Write applications with SWITCH/CASE control structures.

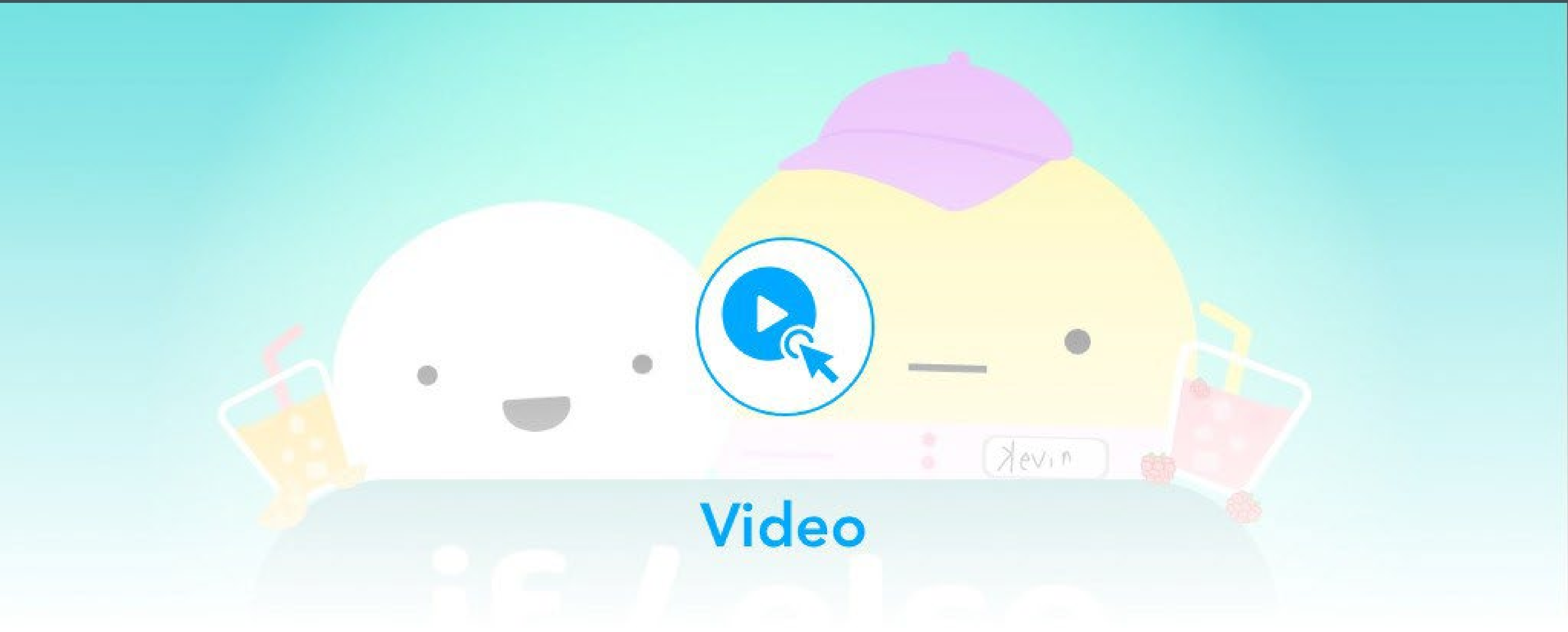
Learning objectives:

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

✓ Write applications in C++ using ‘if’ control structures for both singular and nested types.



Class Activity 1



Coding Basics: If Statements, If Else, Else - Coding Tutorial For Java, C, and C++!

Control flow in C++ programming

- In C++, statements inside the code are generally executed from top to bottom, in the sequentially order. However, it is not always the case. It may require to execute or skip certain set of instructions based on condition or execute a set of statements repeatedly.

- For that purpose, C++ provides control flow statements that serve to specify what need to be done when and under certain conditions.

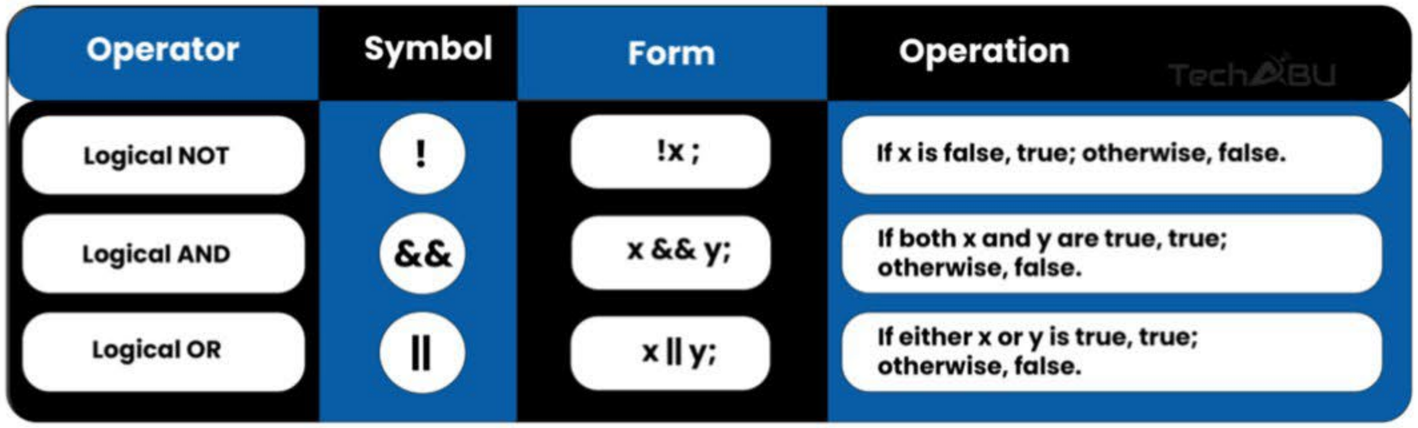
- Control flow is the order in which a program performs actions. Most programming languages use two kinds of statements to regulate control flow.

- Firstly, a branching statement chooses one action from a list of two or more possible actions. Because branching statements choose, or decide, among possible actions, it is also called decision making statements.

- Secondly, a loop statement reiterates an action repeatedly until some stopping condition is met.

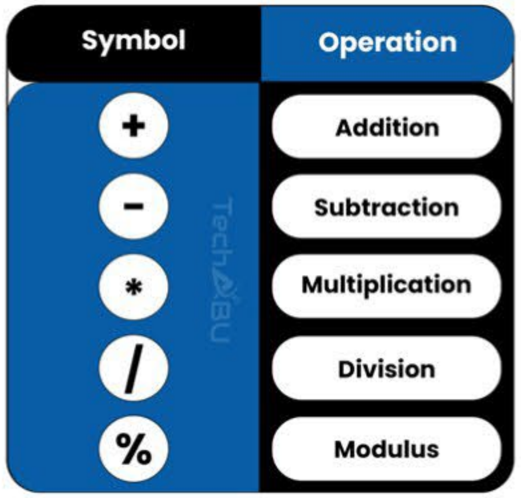
- Control flow statements can utilize (1) Arithmetic Operators, (2) Logical Operators or (3) Comparison Operators for the decision making,

Logical Operators



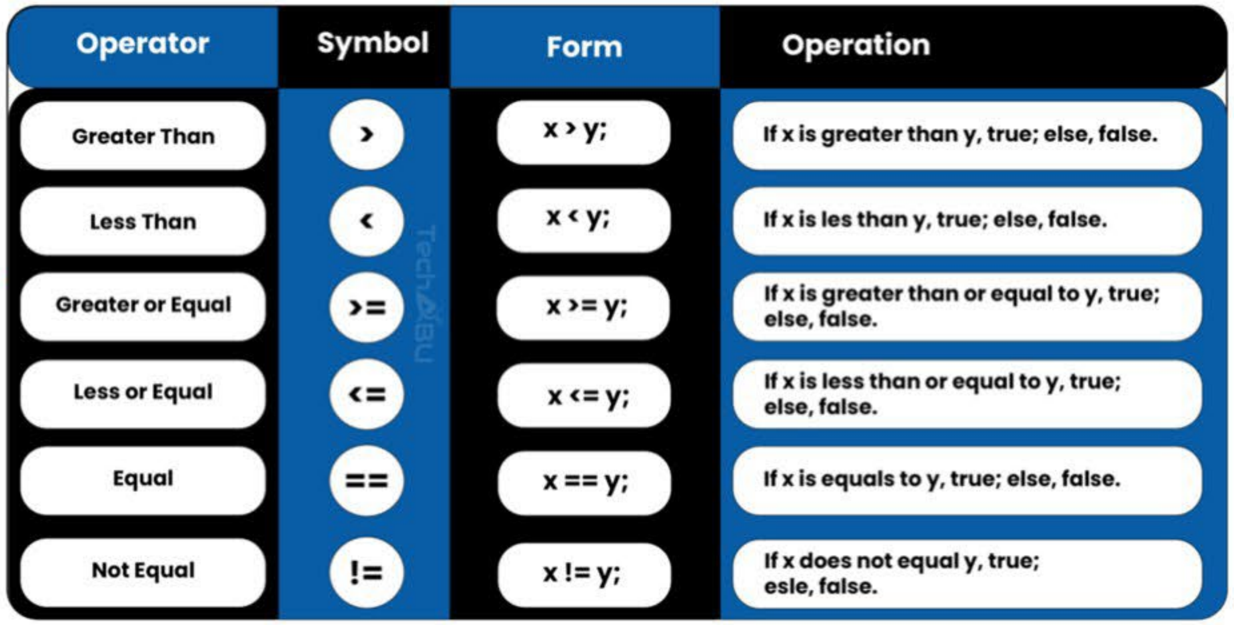
Operator	Symbol	Form	Operation
Logical NOT	!	!x ;	If x is false, true; otherwise, false.
Logical AND	&&	x && y;	If both x and y are true, true; otherwise, false.
Logical OR	||	x || y;	If either x or y is true, true; otherwise, false.

Arithmetic Operators



Symbol	Operation
+	Addition
−	Subtraction
*	Multiplication
/	Division
%	Modulus

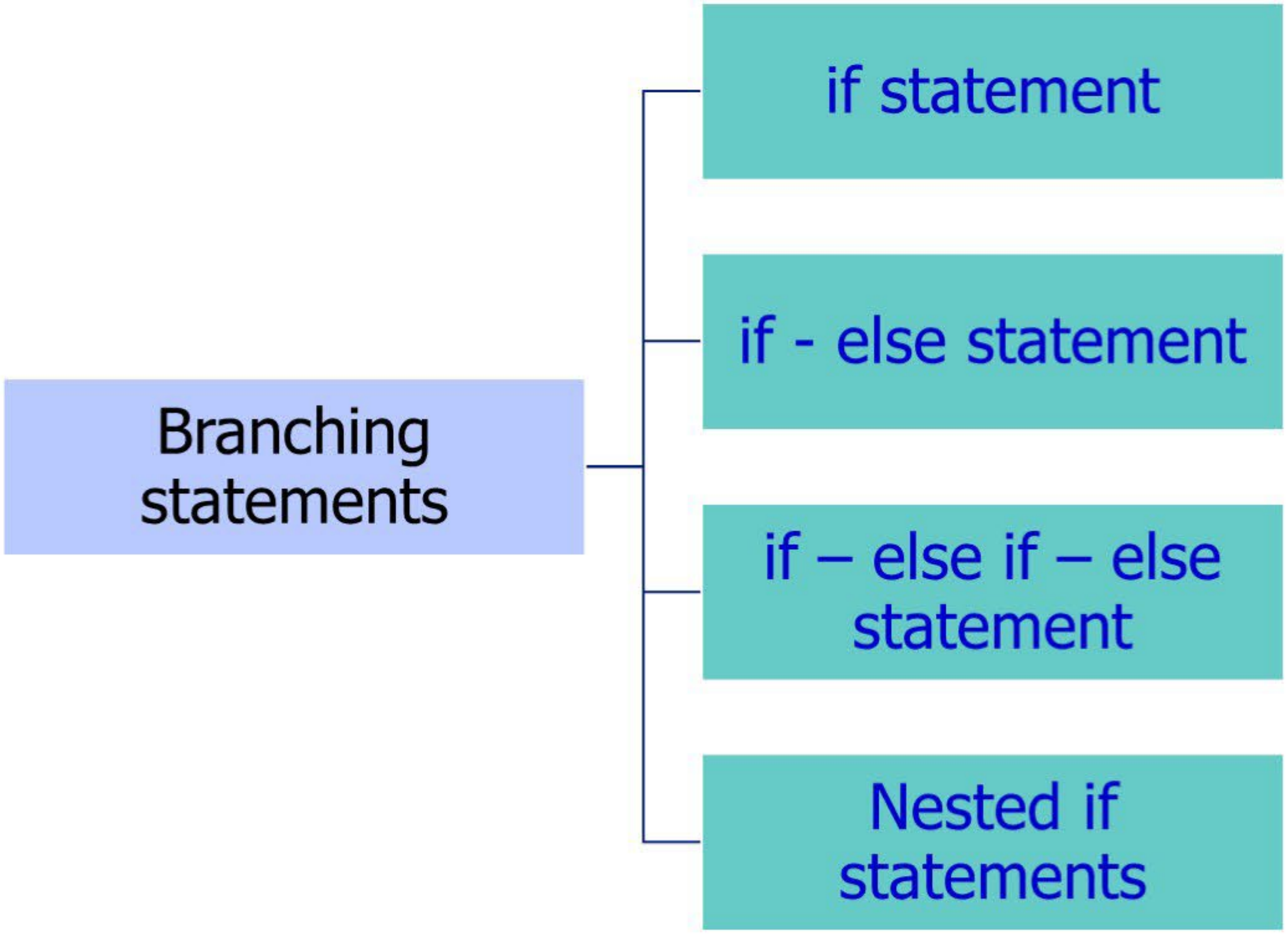
Comparison Operators



Operator	Symbol	Form	Operation
Greater Than	>	x > y;	If x is greater than y, true; else, false.
Less Than	<	x < y;	If x is les than y, true; else, false.
Greater or Equal	>=	x >= y;	If x is greater than or equal to y, true; else, false.
Less or Equal	<=	x <= y;	If x is less than or equal to y, true; else, false.
Equal	==	x == y;	If x is equals to y, true; else, false.
Not Equal	!=	x != y;	If x does not equal y, true; esle, false.

Branching statements

- For the branching statements, there are four types of branching as described below.



Branching statements

if statement

if - else statement

if – else if – else statement

Nested if statements

if statement

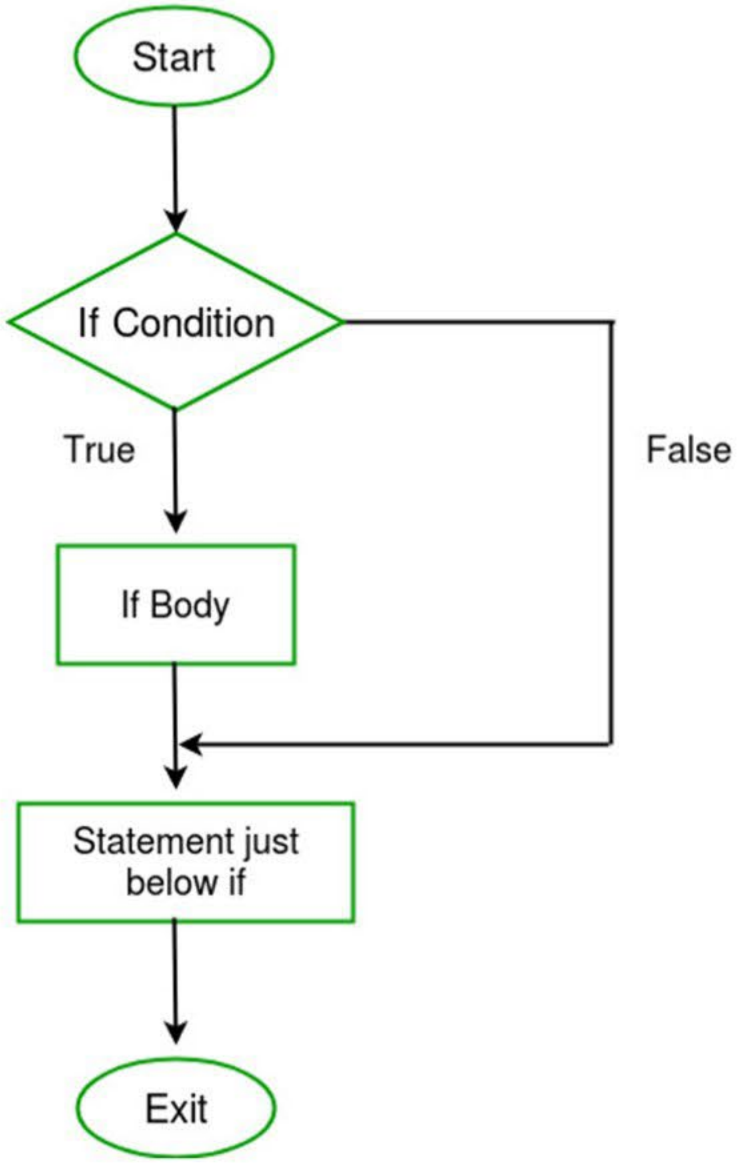
- The **if** keyword is used to execute a statement or block, if, and only if, a condition is fulfilled. Its syntax is:

```
if (condition)
{
// body of if statement
}
```

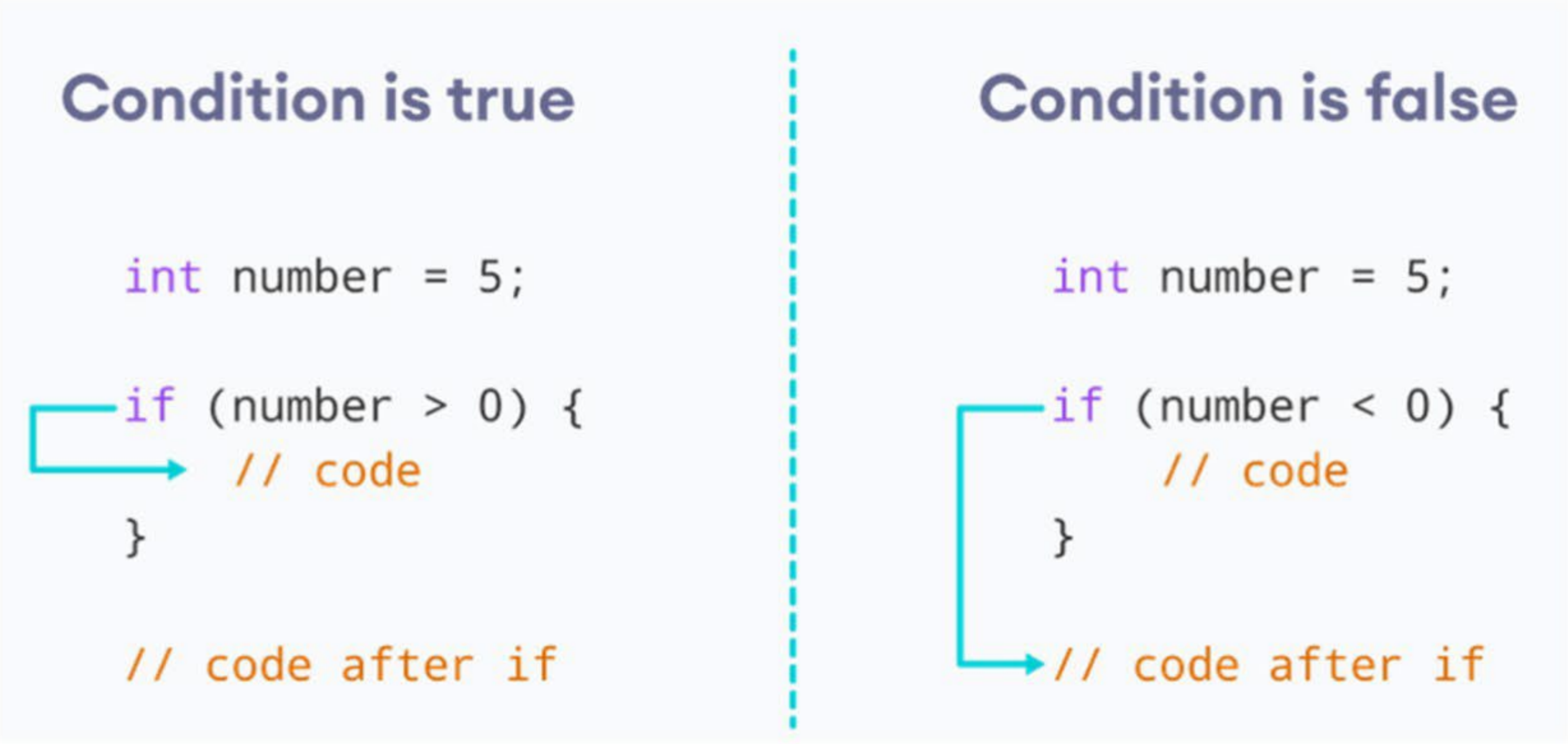
- The if statement evaluates the condition inside the parentheses ( ).

- If the condition evaluates to true, the code inside the body of if is executed. If the condition evaluates to false, the code inside the body of if is skipped.

- The code inside the curly brackets (**{ }**) is the body of the if statement.



- The procedure of if statement is illustrated below. The curly brackets are forming a compound statement.

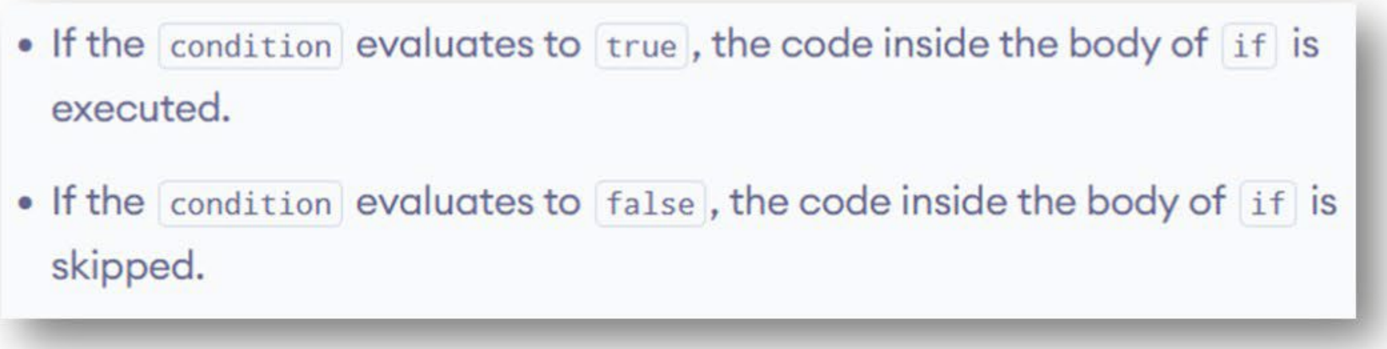


Condition is true

```
int number = 5;

if (number > 0) {
    // code
}

// code after if
```

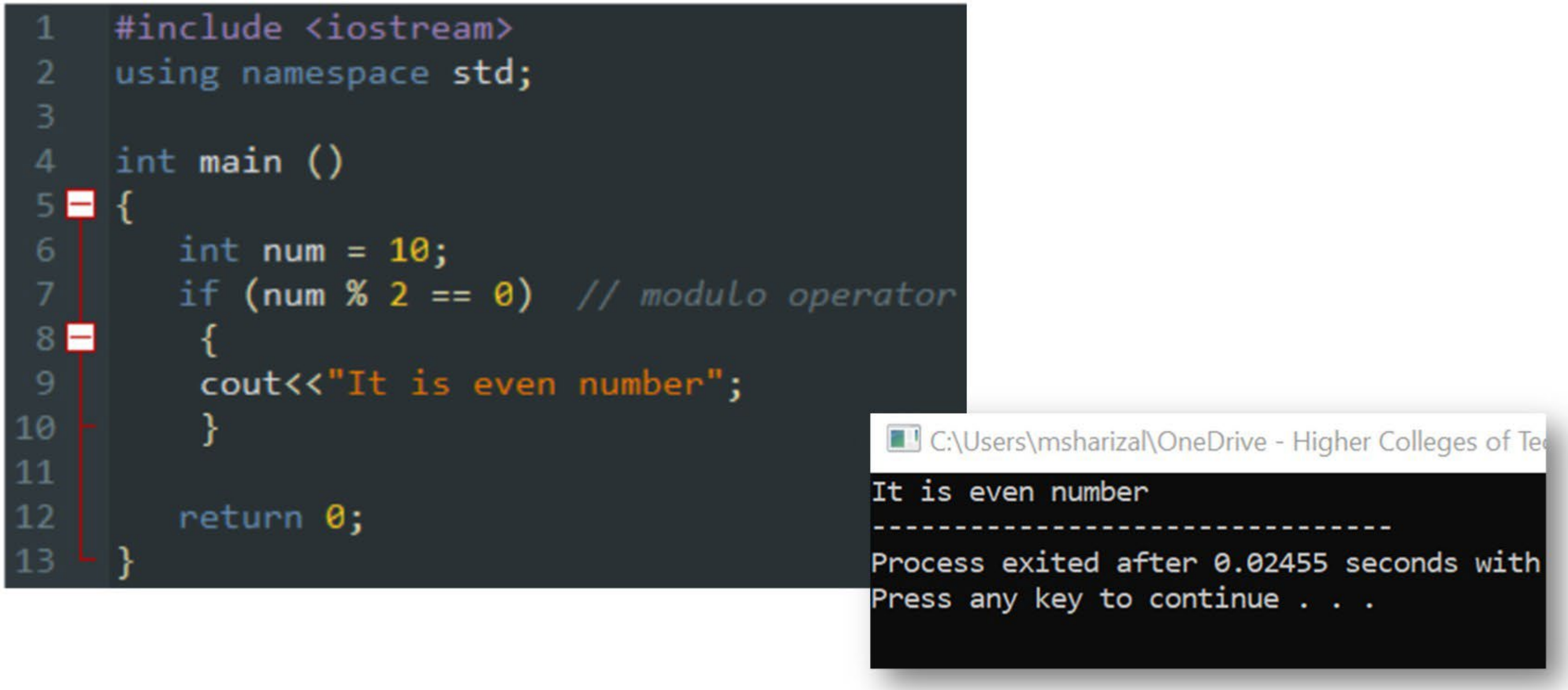
Condition is false

```
int number = 5;

if (number < 0) {
    // code
}

// code after if
```



- If the `condition` evaluates to `true`, the code inside the body of `if` is executed.
- If the `condition` evaluates to `false`, the code inside the body of `if` is skipped.

Example 1 if statement

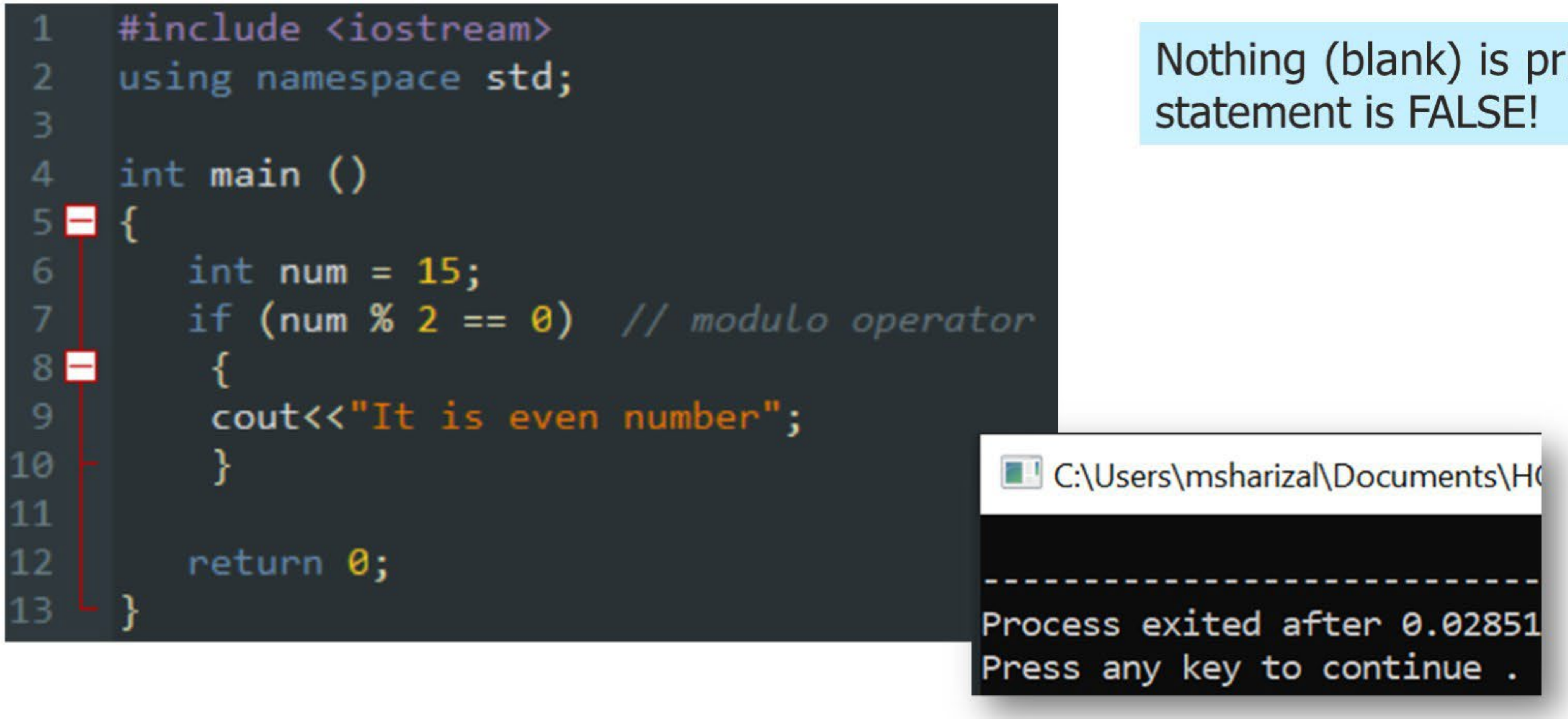


```cpp
#include <iostream>
using namespace std;

int main ()
{
    int num = 10;
    if (num % 2 == 0)  // modulo operator
     {
     cout<<"It is even number";
     }

    return 0;
}
```

C:\Users\msharizal\OneDrive - Higher Colleges of Te

```
It is even number
--------------------------------
Process exited after 0.02455 seconds with
Press any key to continue . . .
```

Example 2 if statement



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

int main ()
{
    int num = 15;
    if (num % 2 == 0)  // modulo operator
     {
     cout<<"It is even number";
     }

    return 0;
}
```

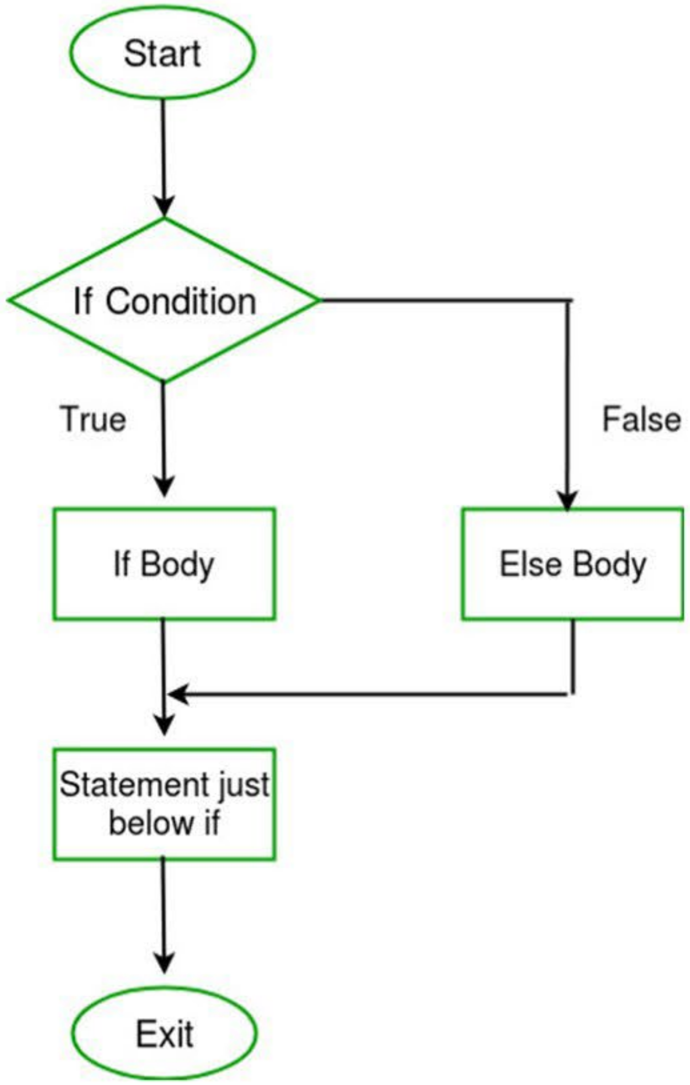
```
C:\Users\msharizal\Documents\H

--------------------------------
Process exited after 0.02851
Press any key to continue .
```

Nothing (blank) is printed-out if statement is FALSE!

if - else statement

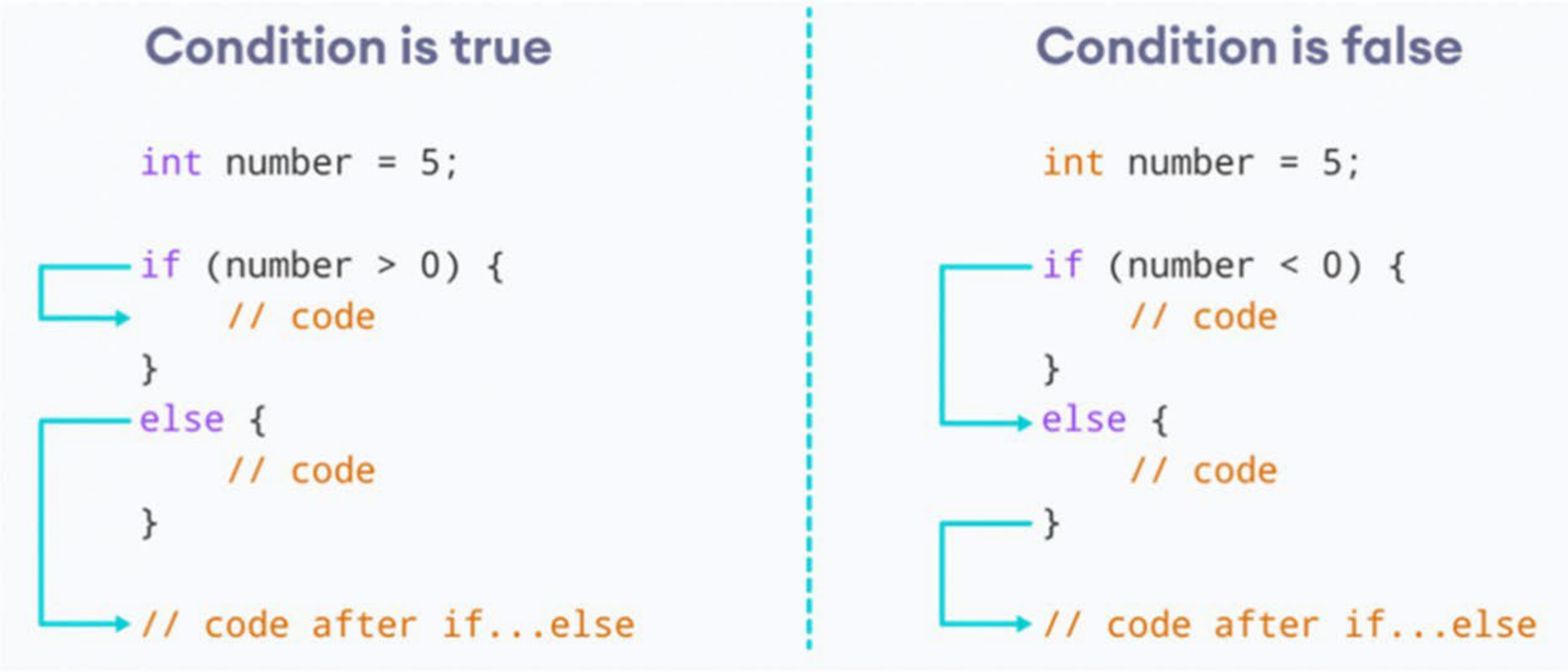
- Selection statements with **if** can also specify what happens when the condition is not fulfilled, by using the **else** keyword to introduce an alternative statement. Its syntax is:

```
if (condition)
{
// block of code OR statement1 if condition is true
}
else
{
// block of code OR statement2  if condition is false
}
```



- The if-else statement evaluates the condition inside the parenthesis. The statement1 is executed in case condition is true, and in case it is not, statement2 is executed.

- The process of if-else statement is illustrated below.

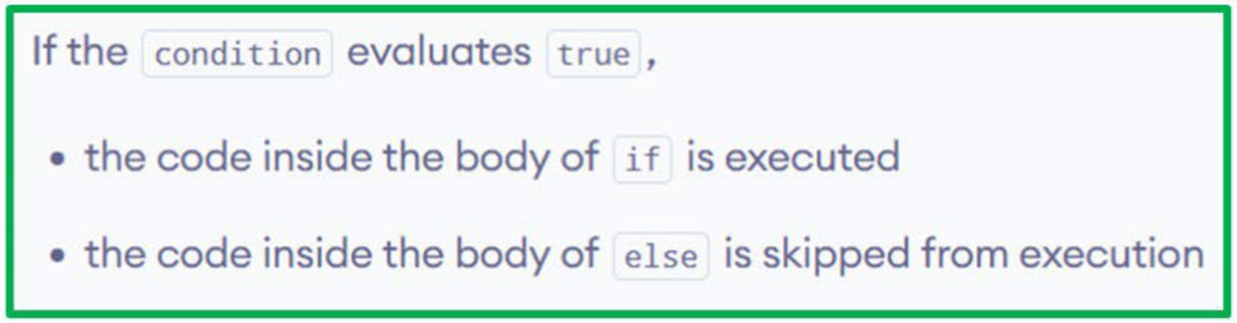


Condition is true

```
int number = 5;

if (number > 0) {
    // code
}
else {
    // code
}

// code after if...else
```

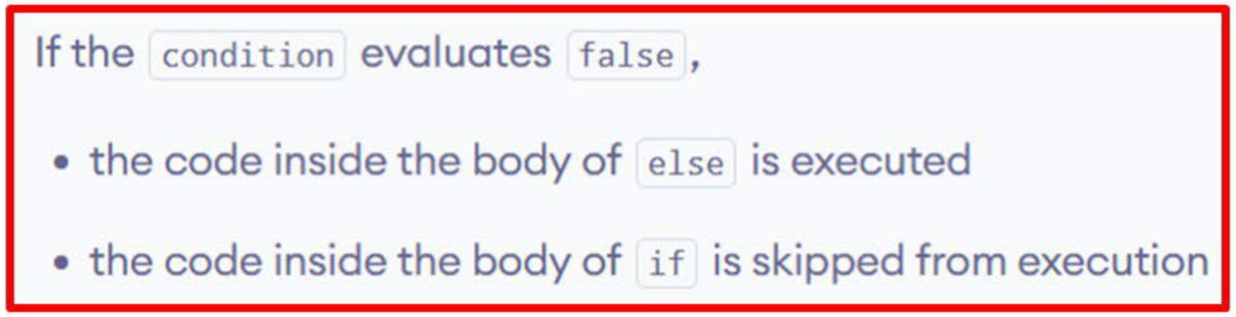
Condition is false

```
int number = 5;

if (number < 0) {
    // code
}
else {
    // code
}

// code after if...else
```



If the `condition` evaluates `true`,

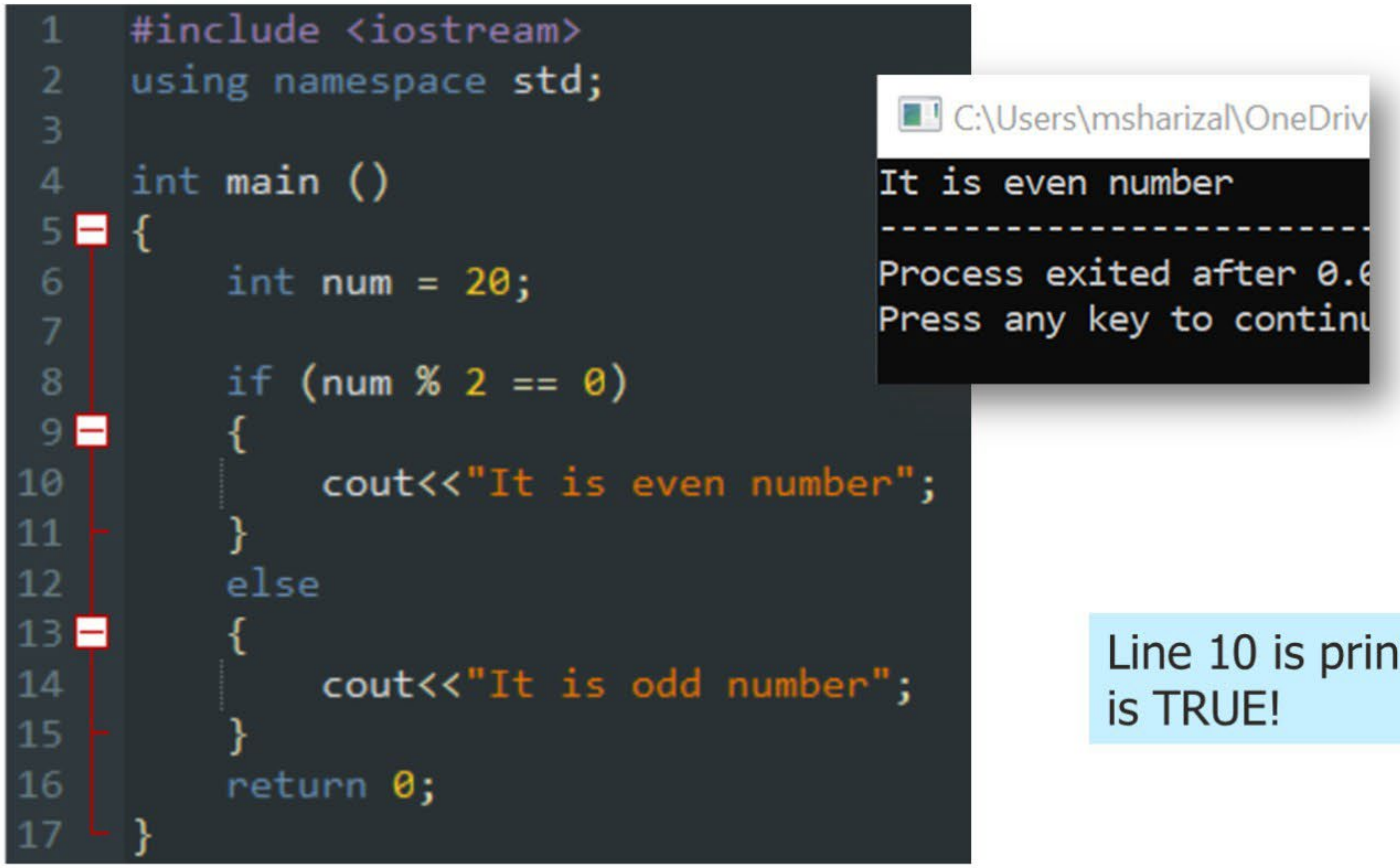
- the code inside the body of `if` is executed
- the code inside the body of `else` is skipped from execution



If the `condition` evaluates `false`,

- the code inside the body of `else` is executed
- the code inside the body of `if` is skipped from execution

Example 3 if – else statement

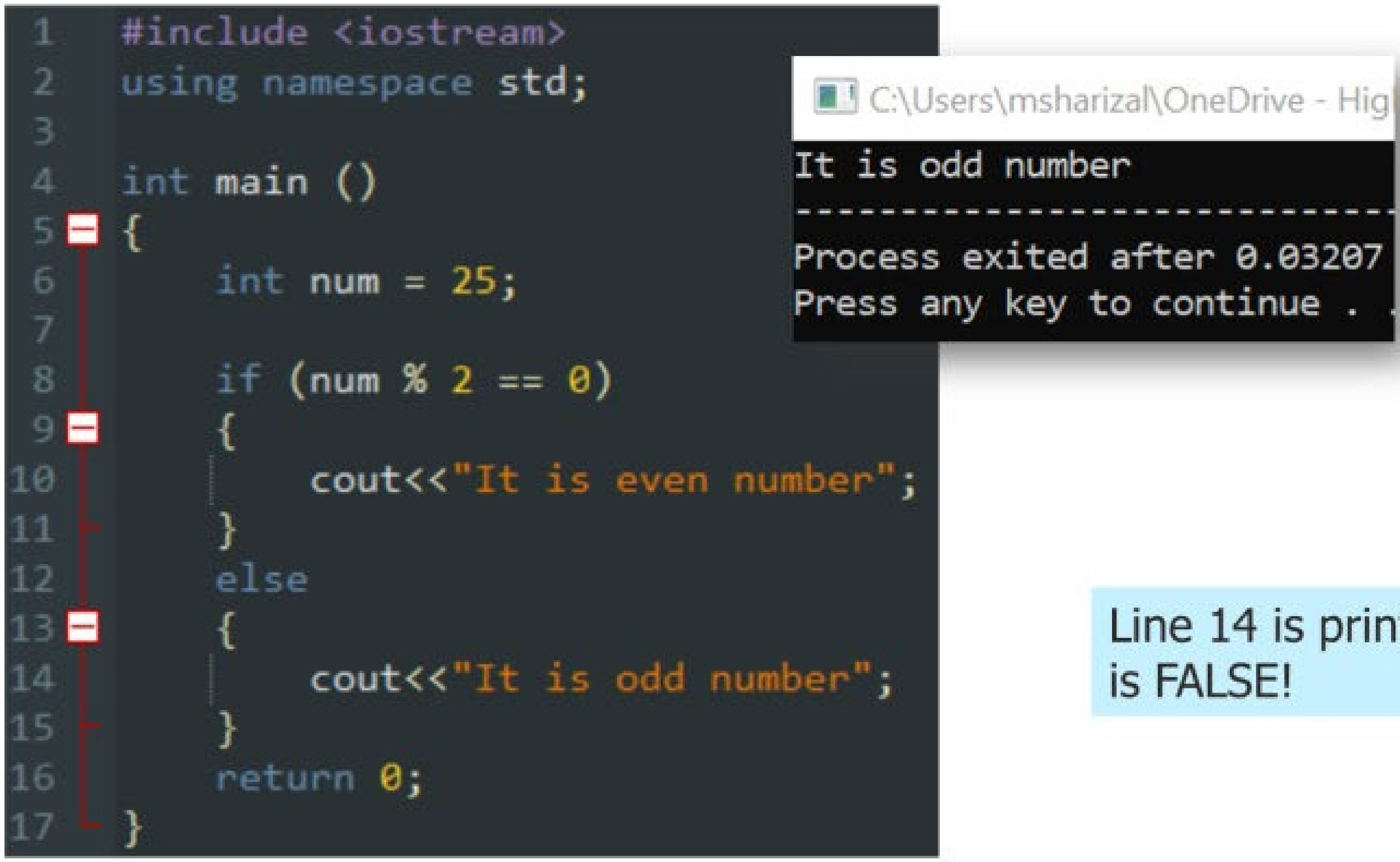


```
1   #include <iostream>
2   using namespace std;
3
4   int main ()
5   {
6       int num = 20;
7
8       if (num % 2 == 0)
9       {
10          cout<<"It is even number";
11      }
12      else
13      {
14          cout<<"It is odd number";
15      }
16      return 0;
17  }
```

```
It is even number
------------------------
Process exited after 0.6
Press any key to continu
```

Line 10 is printed-out if statement is TRUE!

Example 4 if – else statement



```
1   #include <iostream>
2   using namespace std;
3
4   int main ()
5   {
6       int num = 25;
7
8       if (num % 2 == 0)
9       {
10          cout<<"It is even number";
11      }
12      else
13      {
14          cout<<"It is odd number";
15      }
16      return 0;
17  }
```

C:\Users\msharizal\OneDrive - Hig

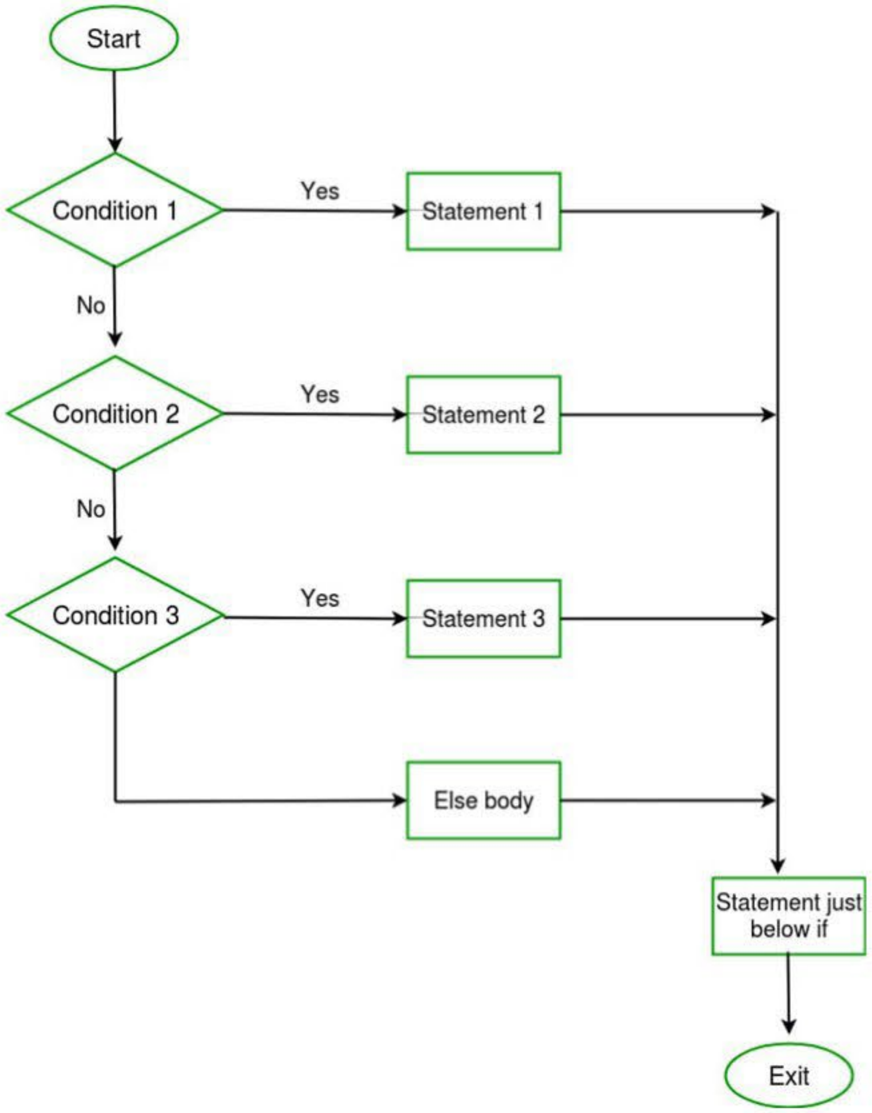
```
It is odd number
-----------------------------
Process exited after 0.03207
Press any key to continue . .
```

Line 14 is printed-out if statement is FALSE!

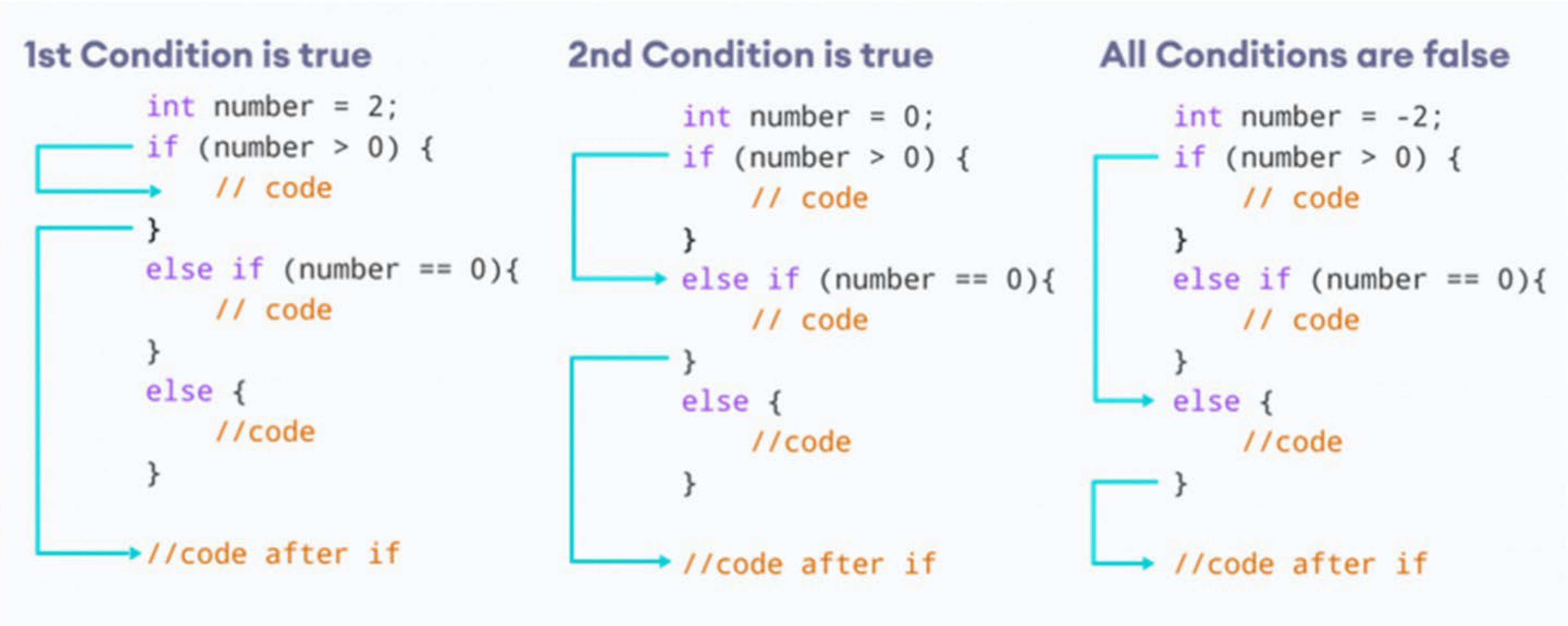
if - else if - else statement

- The if-else statement is used to execute a block of code among 2 alternatives. If we need to make a choice for more than 2 alternatives, we use the if-else if-else statement. The syntax is:

```
if (condition1)
{
// code block 1
}
else if (condition2)
{
// code block 2
}
else
{
// code block 3
}
```



- The execution of if-else if-else statement is illustrated below. **There can be more than one else if statement but only one if and else statements**.



1st Condition is true

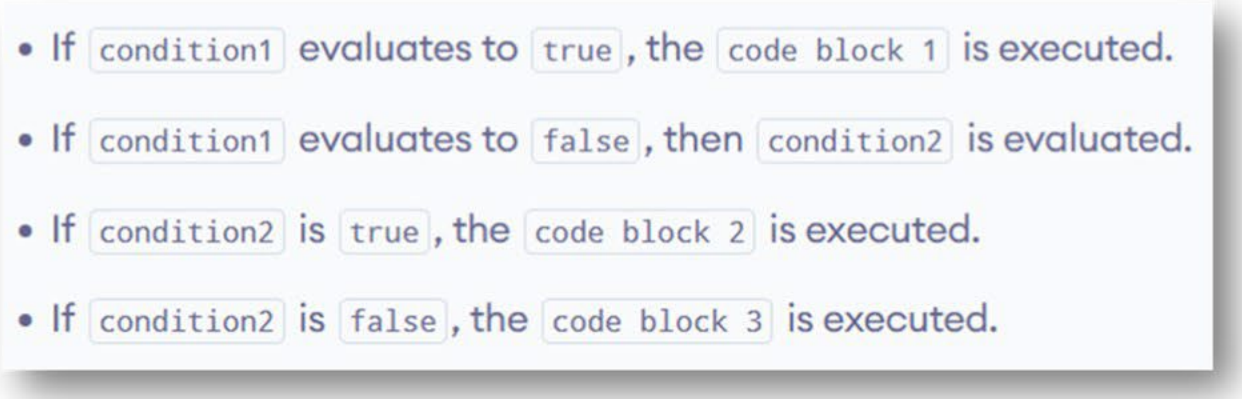
```
int number = 2;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

2nd Condition is true

```
int number = 0;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

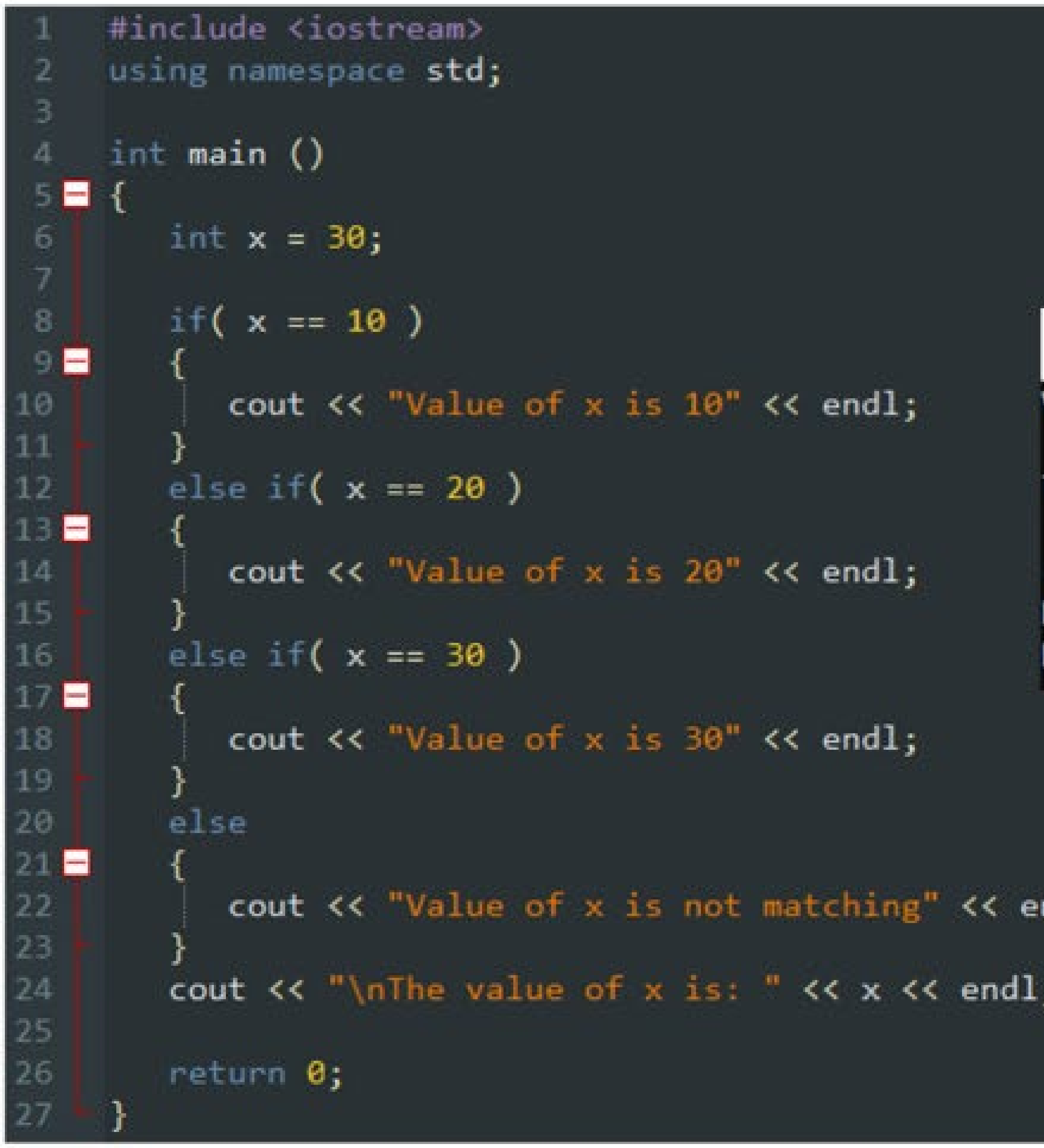
All Conditions are false

```
int number = -2;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

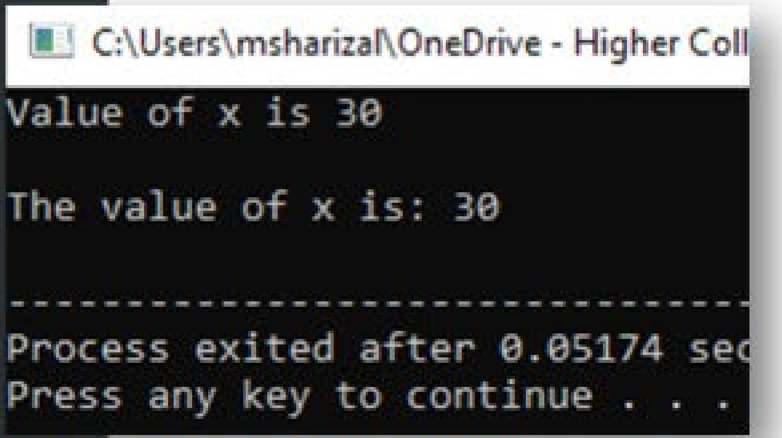


- If `condition1` evaluates to `true`, the `code block 1` is executed.
- If `condition1` evaluates to `false`, then `condition2` is evaluated.
- If `condition2` is `true`, the `code block 2` is executed.
- If `condition2` is `false`, the `code block 3` is executed.

Example 5 if – else if - else statement



```
1   #include <iostream>
2   using namespace std;
3
4   int main ()
5   {
6       int x = 30;
7
8       if( x == 10 )
9       {
10          cout << "Value of x is 10" << endl;
11      }
12      else if( x == 20 )
13      {
14          cout << "Value of x is 20" << endl;
15      }
16      else if( x == 30 )
17      {
18          cout << "Value of x is 30" << endl;
19      }
20      else
21      {
22          cout << "Value of x is not matching" << endl;
23      }
24      cout << "\nThe value of x is: " << x << endl;
25
26      return 0;
27  }
```

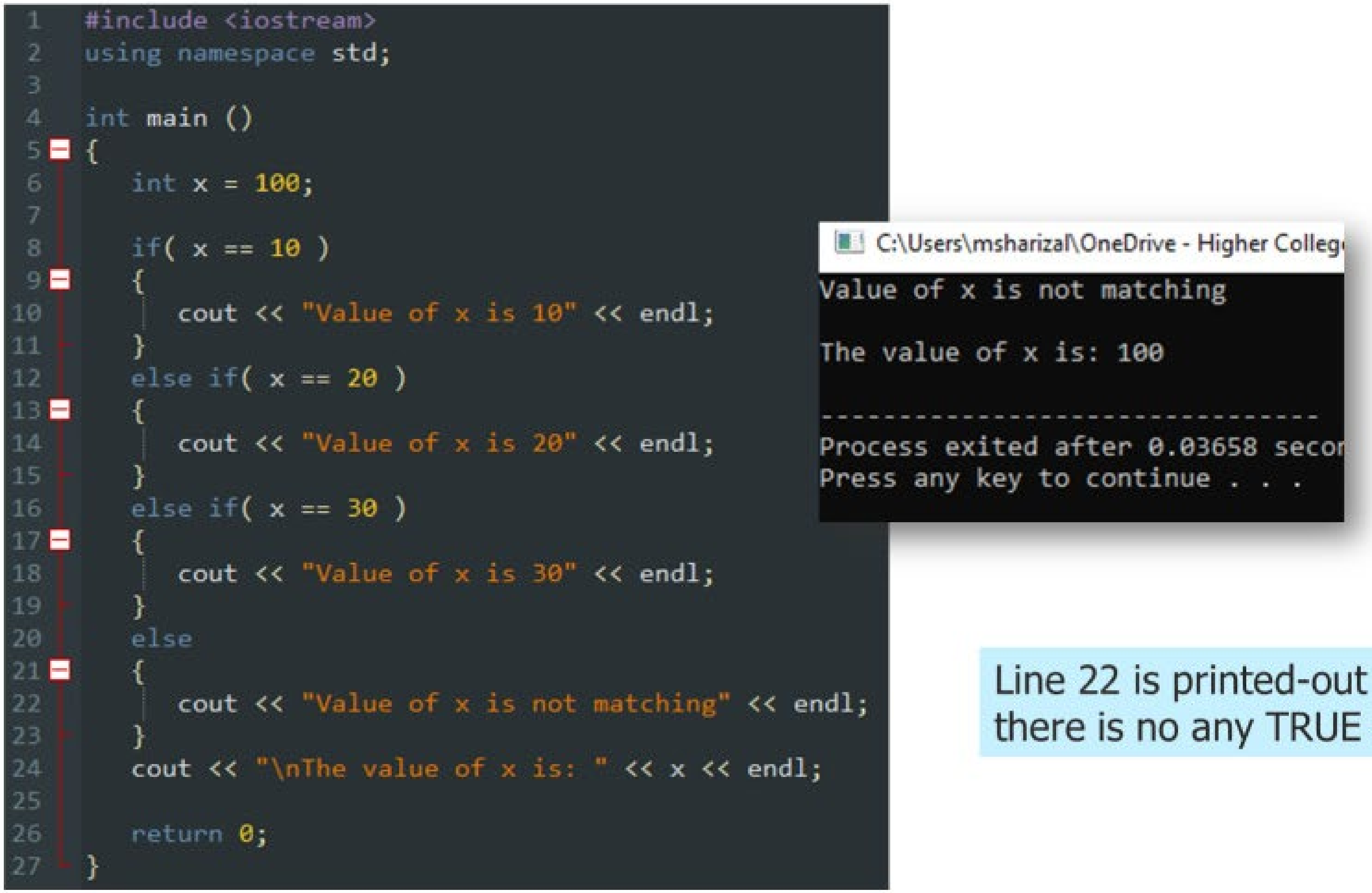


C:\Users\msharizal\OneDrive - Higher Coll

```
Value of x is 30

The value of x is: 30

--------------------------------
Process exited after 0.05174 sec
Press any key to continue . . .
```

Line 18 is printed-out for since it is a TRUE statement!

Example 6 if – else if - else statement



```
1   #include <iostream>
2   using namespace std;
3
4   int main ()
5   {
6       int x = 100;
7
8       if( x == 10 )
9       {
10          cout << "Value of x is 10" << endl;
11      }
12      else if( x == 20 )
13      {
14          cout << "Value of x is 20" << endl;
15      }
16      else if( x == 30 )
17      {
18          cout << "Value of x is 30" << endl;
19      }
20      else
21      {
22          cout << "Value of x is not matching" << endl;
23      }
24      cout << "\nThe value of x is: " << x << endl;
25
26      return 0;
27  }
```

C:\Users\msharizal\OneDrive - Higher Colleg

```
Value of x is not matching

The value of x is: 100

--------------------------------
Process exited after 0.03658 secon
Press any key to continue . . .
```

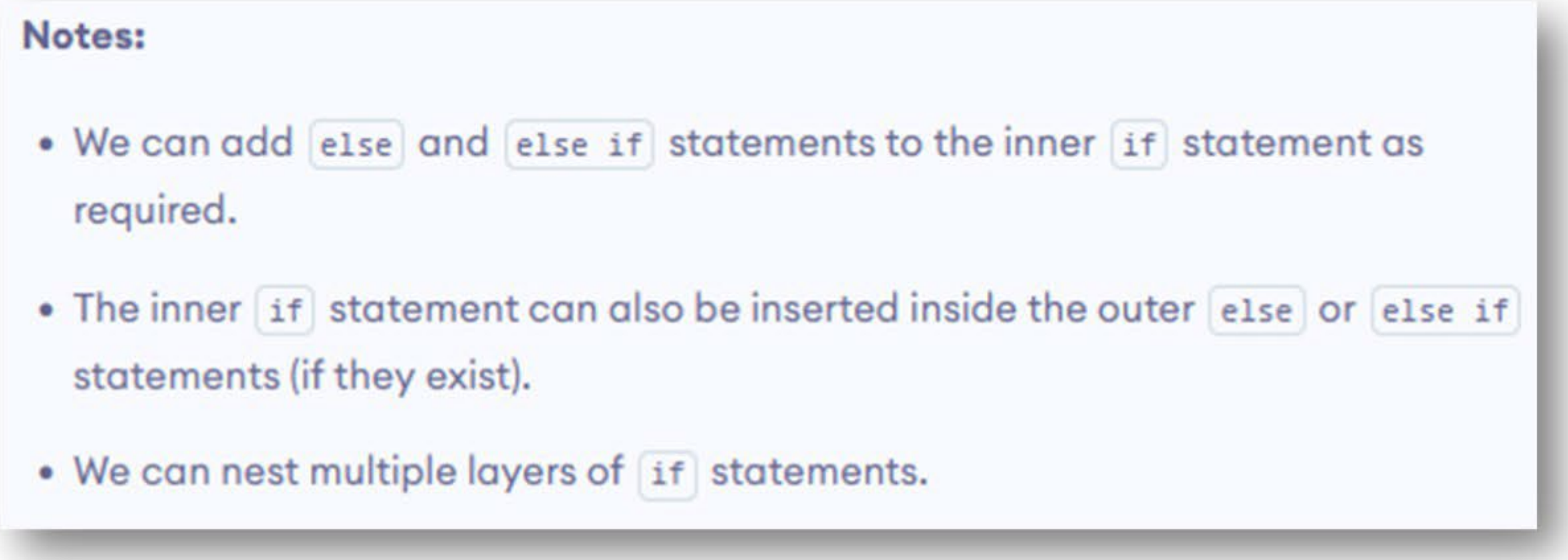
Line 22 is printed-out for since there is no any TRUE statement!

Nested if statements

- Sometimes in problem solving the computer program, we need to use an if statement inside another if statement. This is known as nested if statement.

- Think of it as multiple layers of if statements. There is a first, outer if statement, and inside it is another, inner if statement. Its syntax is:

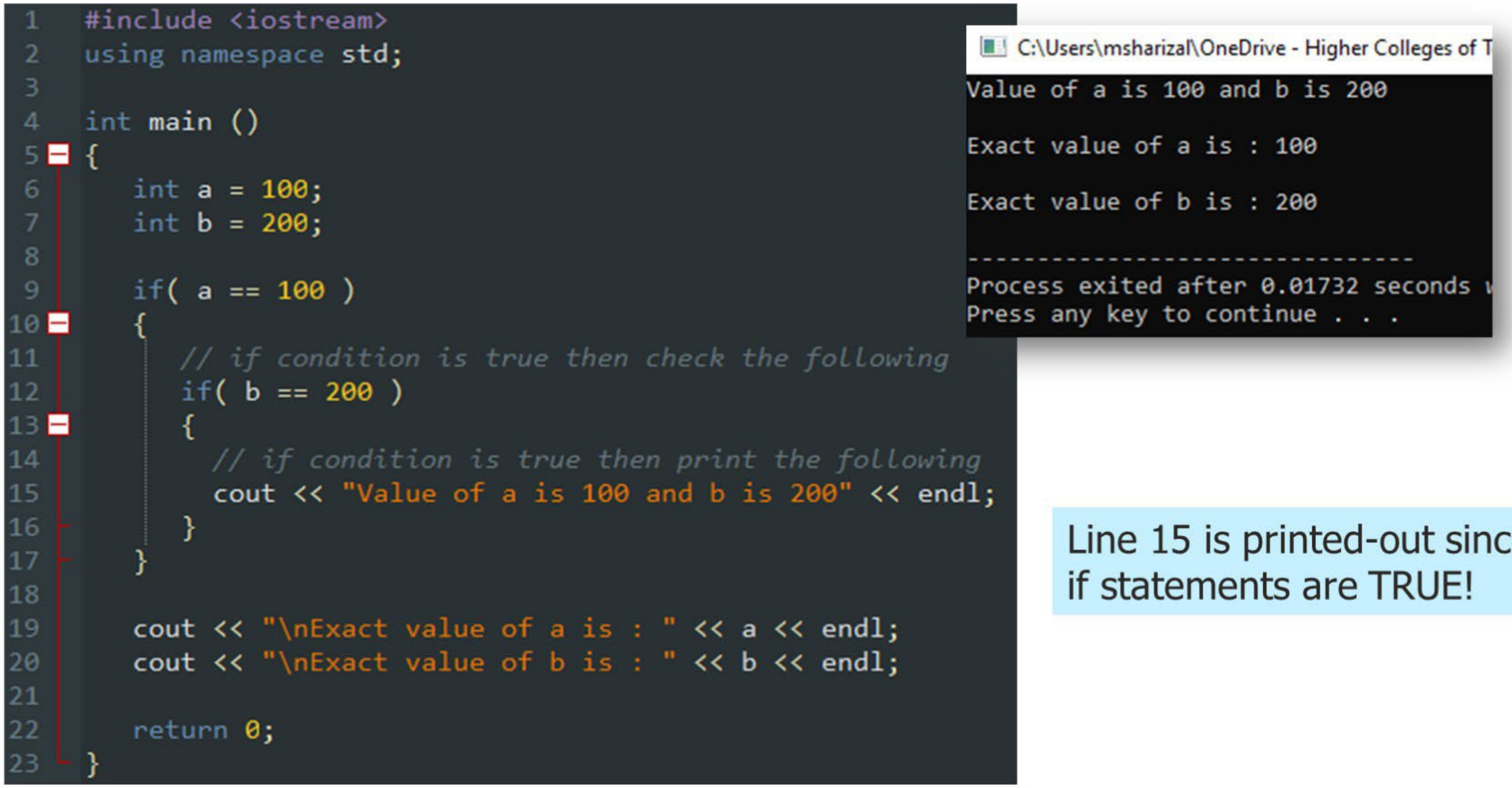
```
// outer if statement
if (condition1)
{
// statements
     // inner if statement
     if (condition2)
     {
        // statements
      }
  }
```



Notes:

- We can add `else` and `else if` statements to the inner `if` statement as required.
- The inner `if` statement can also be inserted inside the outer `else` or `else if` statements (if they exist).
- We can nest multiple layers of `if` statements.

Example 7 Nested if statements



```
1   #include <iostream>
2   using namespace std;
3
4   int main ()
5   {
6       int a = 100;
7       int b = 200;
8
9       if( a == 100 )
10      {
11          // if condition is true then check the following
12          if( b == 200 )
13          {
14            // if condition is true then print the following
15            cout << "Value of a is 100 and b is 200" << endl;
16          }
17      }
18
19      cout << "\nExact value of a is : " << a << endl;
20      cout << "\nExact value of b is : " << b << endl;
21
22      return 0;
23  }
```

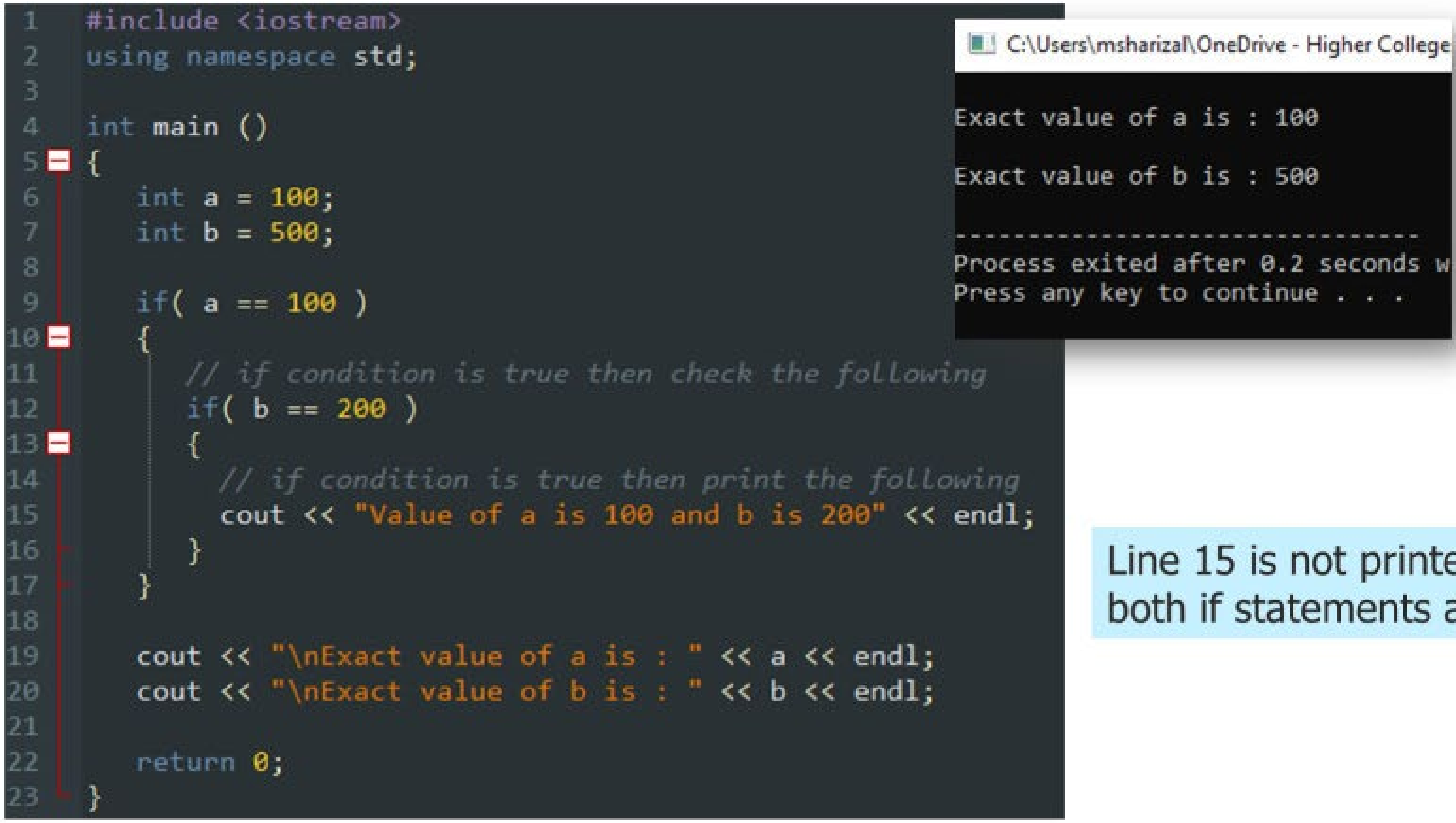
C:\Users\msharizal\OneDrive - Higher Colleges of T

```
Value of a is 100 and b is 200

Exact value of a is : 100

Exact value of b is : 200

--------------------------------
Process exited after 0.01732 seconds w
Press any key to continue . . .
```

Line 15 is printed-out since both if statements are TRUE!

Example 8

Nested if statements



```
1  #include <iostream>
2  using namespace std;
3
4  int main ()
5  {
6      int a = 100;
7      int b = 500;
8
9      if( a == 100 )
10     {
11         // if condition is true then check the following
12         if( b == 200 )
13         {
14           // if condition is true then print the following
15           cout << "Value of a is 100 and b is 200" << endl;
16         }
17     }
18
19     cout << "\nExact value of a is : " << a << endl;
20     cout << "\nExact value of b is : " << b << endl;
21
22     return 0;
23 }
```

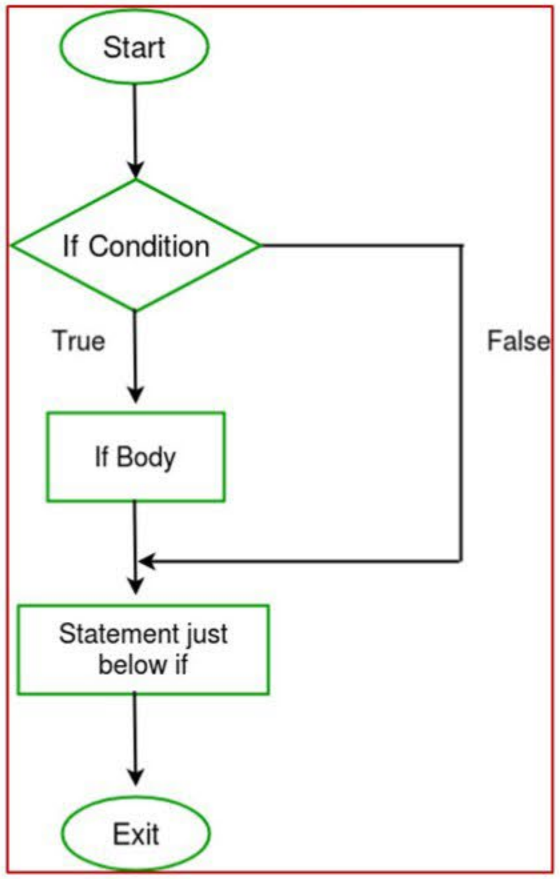
C:\Users\msharizal\OneDrive - Higher College

```

Exact value of a is : 100

Exact value of b is : 500

--------------------------------
Process exited after 0.2 seconds w
Press any key to continue . . .
```

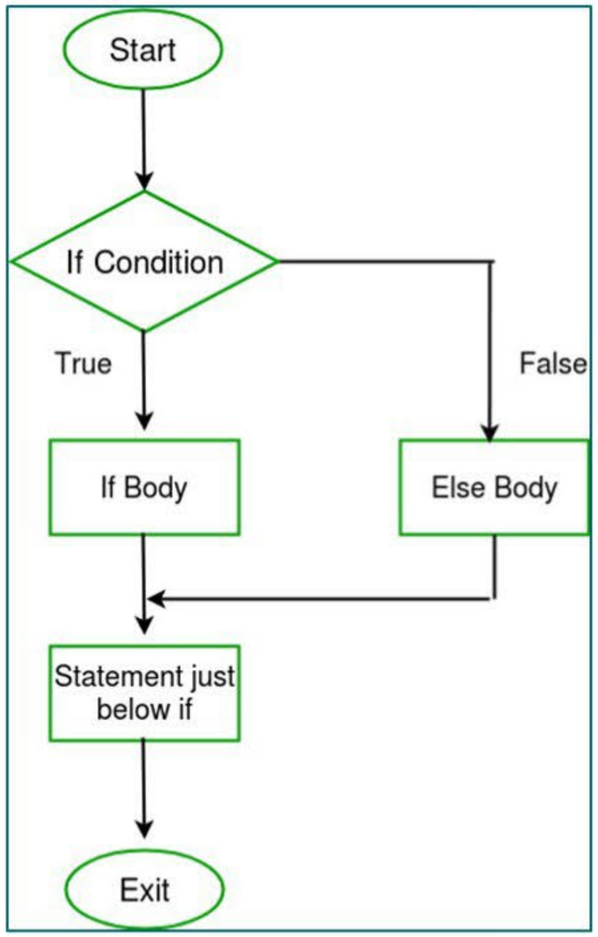
Line 15 is not printed-out since both if statements are FALSE!

Comparison from flowchart perspective

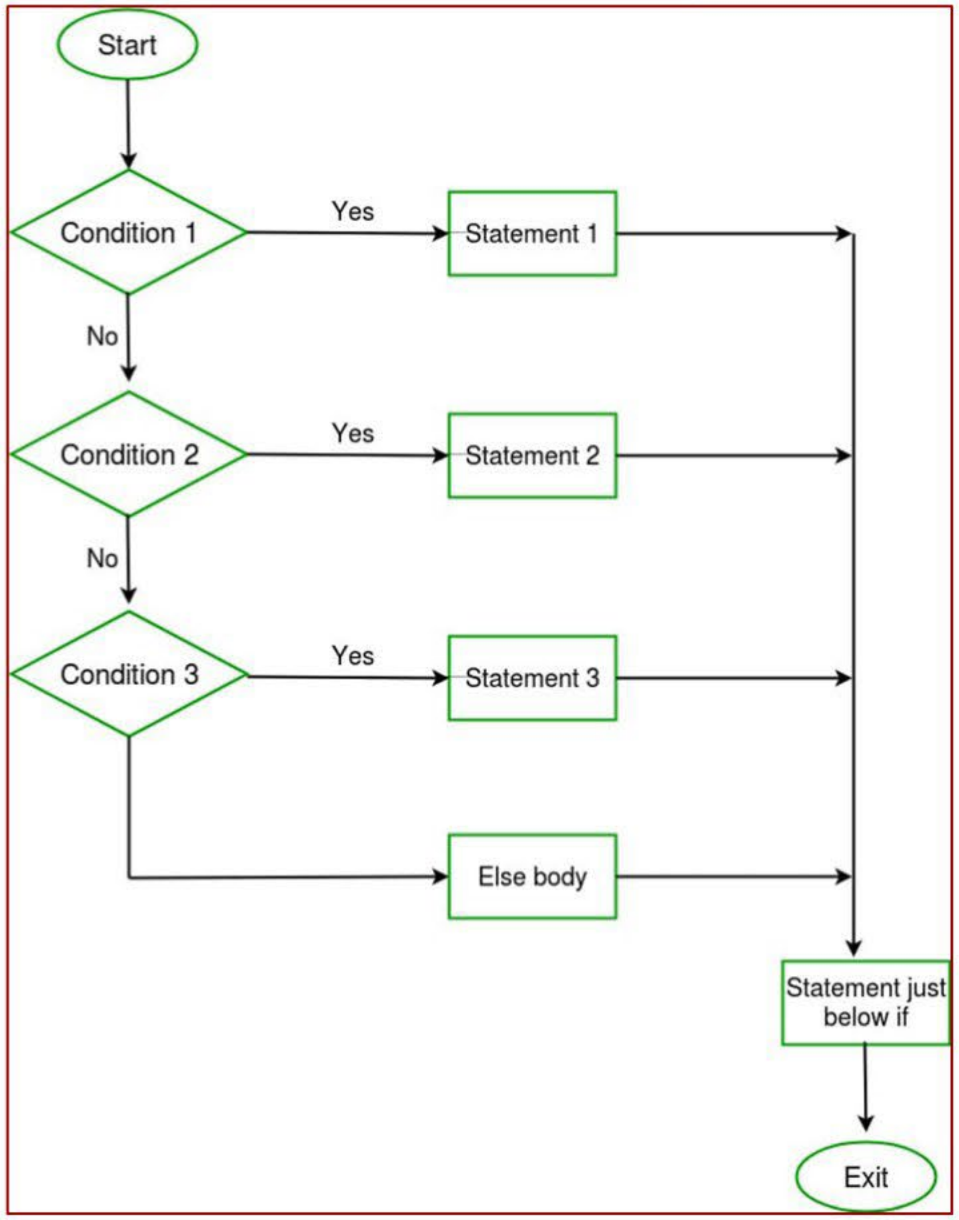
if statement



if - else statement

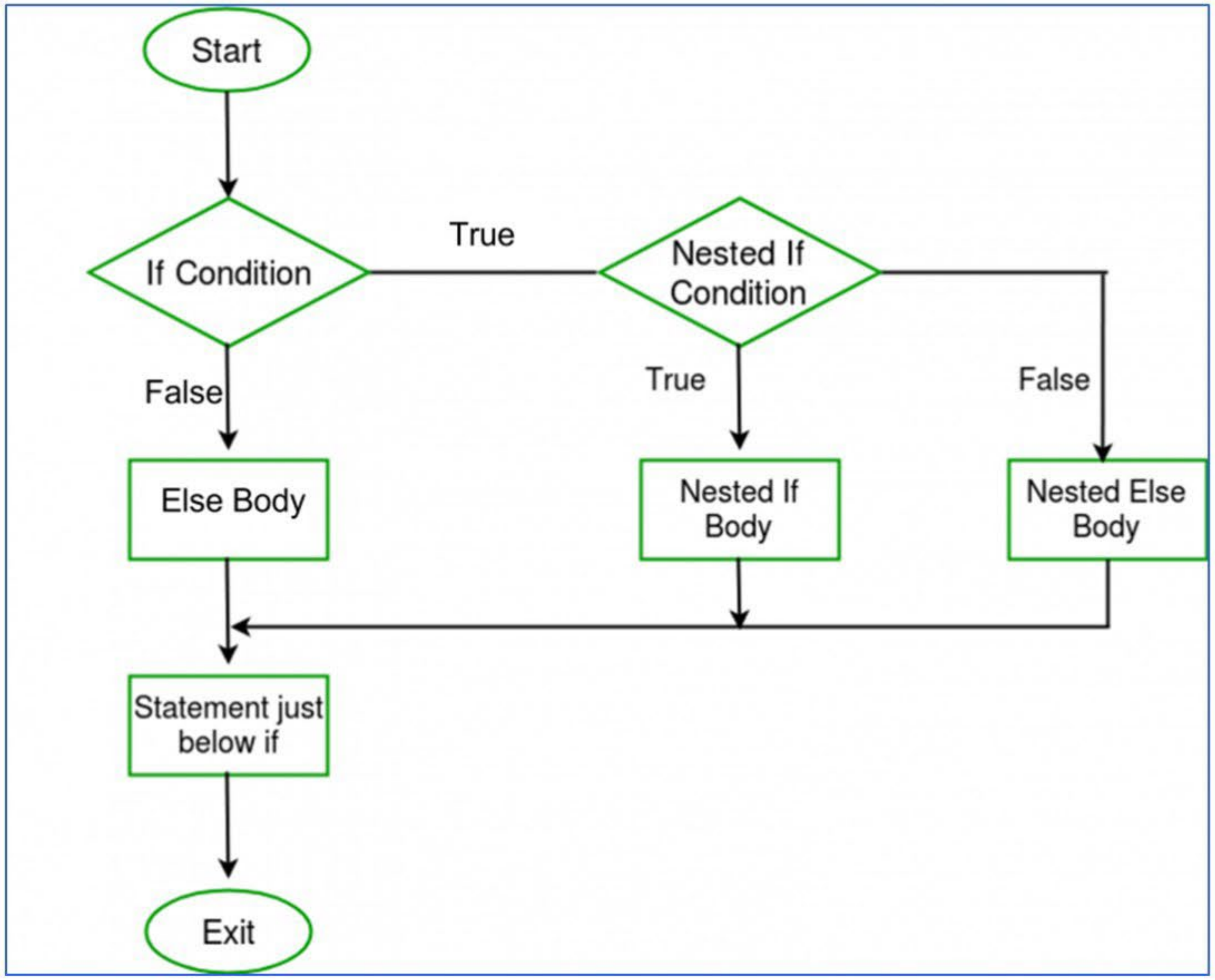


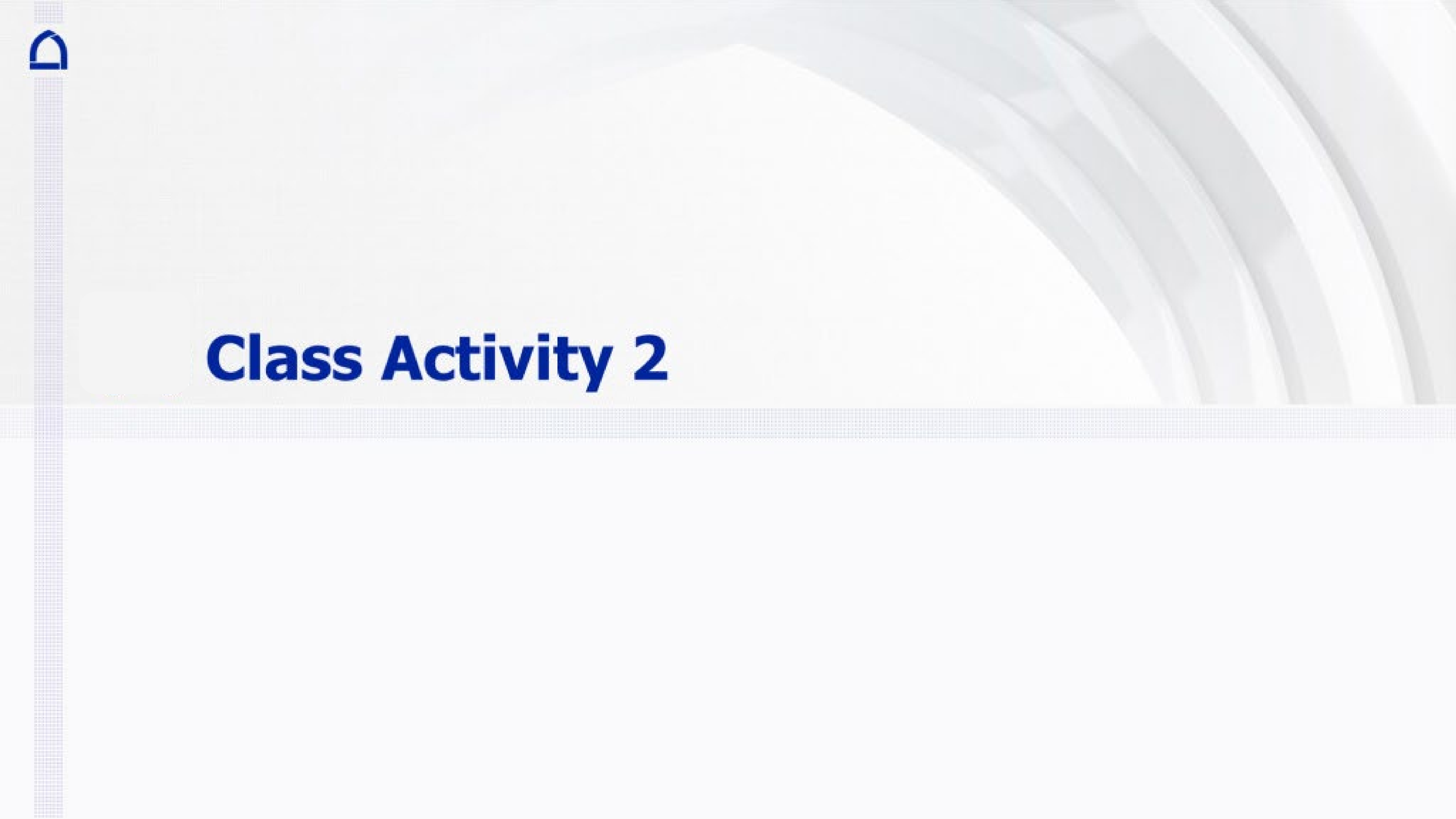
if - else if – else statement



Comparison from flowchart perspective

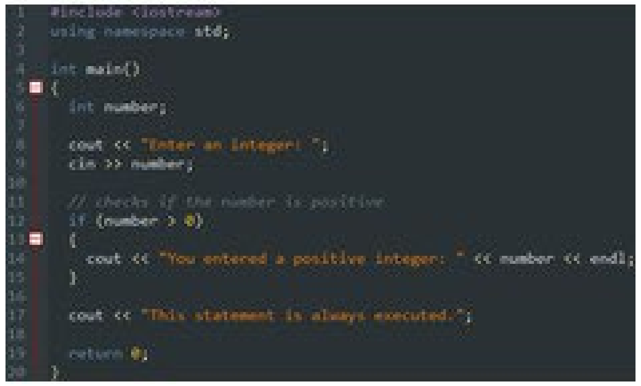
Nested statement





Class Activity 2

Quiz



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

int main()
{
  int number;

  cout << "Enter an integer: ";
  cin >> number;

  // checks if the number is positive
  if (number > 0)
  {
    cout << "You entered a positive integer: " << number << endl;
  }

  cout << "This statement is always executed.";

  return 0;
}
```

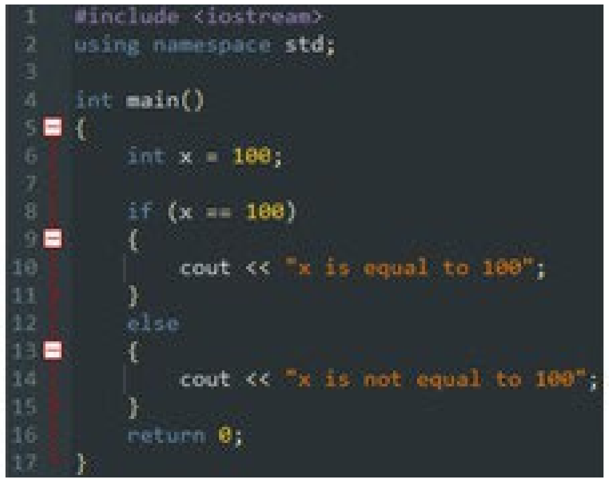
The C++ program shown in the image is representing ____________.

○ **nested if statement**

○ if-else if-else statement

○ if-else statement

○ **if statement**



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

int main()
{
    int x = 100;

    if (x == 100)
    {
        cout << "x is equal to 100";
    }
    else
    {
        cout << "x is not equal to 100";
    }
    return 0;
}
```

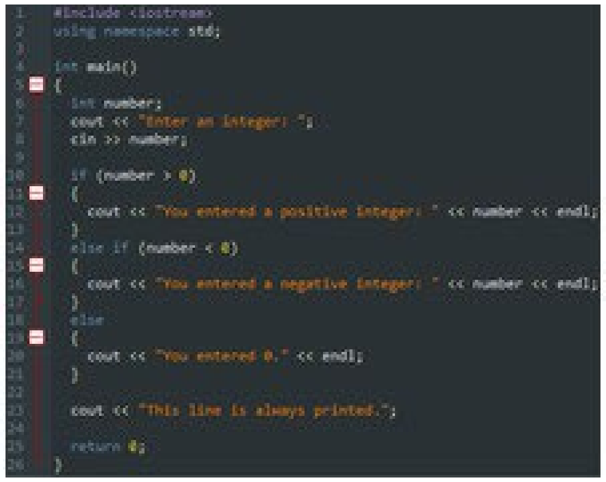
The C++ program shown in the image is representing ___________.

○ **nested if statement**

○ if-else if-else statement

○ if-else statement

○ **if statement**



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

int main()
{
  int number;
  cout << "Enter an integer: ";
  cin >> number;

  if (number > 0)
  {
    cout << "You entered a positive integer: " << number << endl;
  }
  else if (number < 0)
  {
    cout << "You entered a negative integer: " << number << endl;
  }
  else
  {
    cout << "You entered 0." << endl;
  }

  cout << "This line is always printed.";

  return 0;
}
```

The C++ program shown in the image is representing ___________.

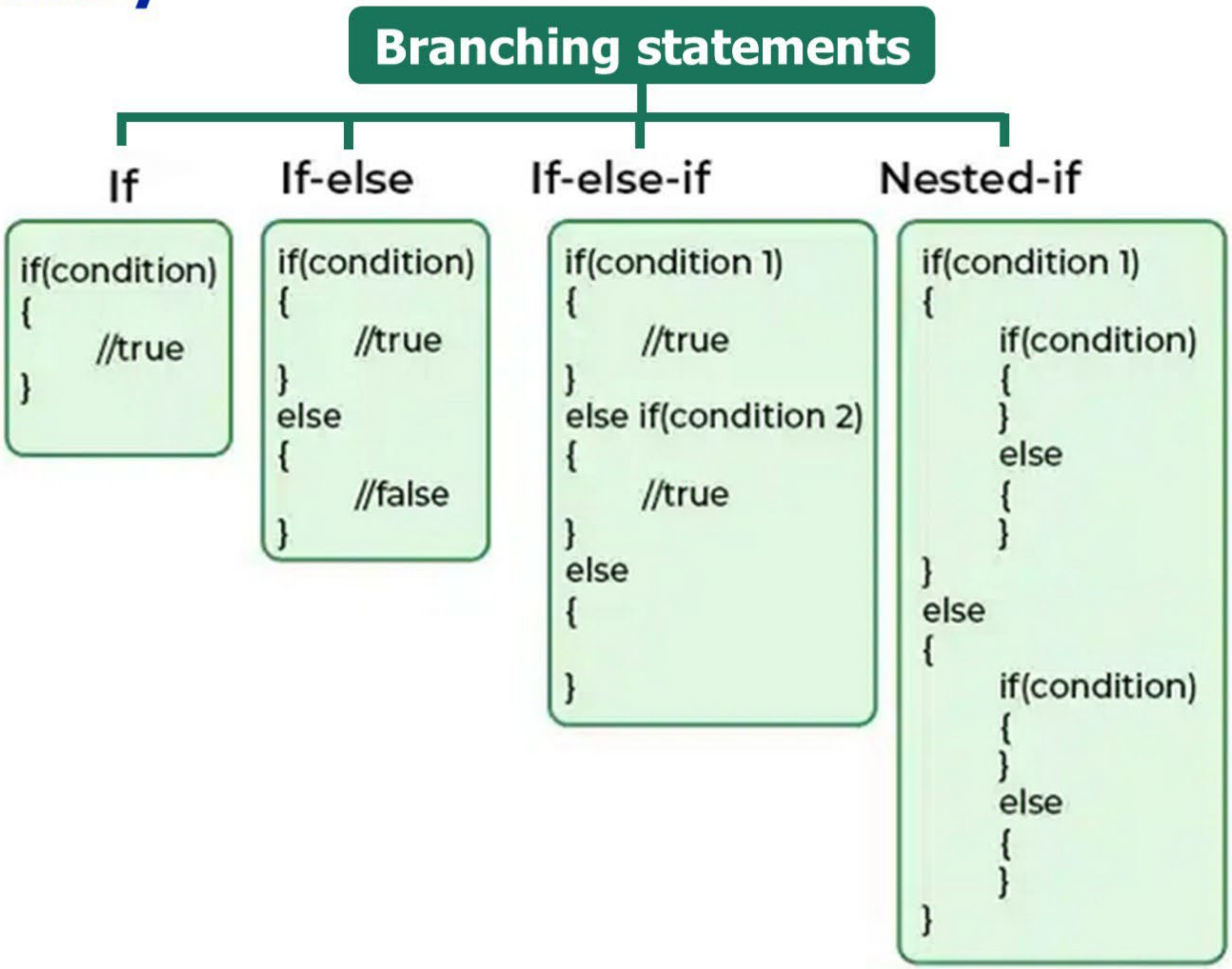
○ **nested if statement**

○ if-else if-else statement

○ if-else statement

○ **if statement**

Summary



Branching statements

If

```
if(condition)
{
        //true
}
```

If-else

```
if(condition)
{
        //true
}
else
{
        //false
}
```

If-else-if

```
if(condition 1)
{
        //true
}
else if(condition 2)
{
        //true
}
else
{

}
```

Nested-if

```
if(condition 1)
{
        if(condition)
        {
        }
        else
        {
        }
}
else
{
        if(condition)
        {
        }
        else
        {
        }
}
```

Higher Colleges of Technology

كليات التقنية العليا

Thank You

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