



CC-301 Programming Fundamentals

Lecture 5

Engr. Waseem Ullah Khan



Selection / Conditional Statements

Used to make decisions based on a given condition

Have two states *true or false*

The true state in programming is 1 and the false state is 0

Example:

Marks: Usually we consider marks above 35 to have passed and below 35 to have failed.

Here we have the choice of pass or fail, which is decided based on marks scored

Offers: If you purchase 4000 and above, 10% discount else no discount. Again, here we have the choice of whether to do the shopping for more than 4000 to be eligible for a 10% discount.



Selection / Conditional Statements

To run one block of code under certain conditions and another block of code under different conditions

For example, assigning grades (A, B, C) based on marks obtained by a student

- if the percentage is above 90, assign grade A
- if the percentage is above 75, assign grade B
- if the percentage is above 65, assign grade C

There are three forms of if...else statements in C++

- if statement
- if...else statement
- if...else if...else statement



C++ if Statement

Used where the execution of an action depends on the satisfaction of a condition

- if condition is true \longrightarrow action is done
- if condition is false \longrightarrow action is not done

Example:

If Ali's height is greater than 6 feet

Ali can become a member of the Basket Ball team

If Ali's height is less than 6 feet

Ali cannot become a member of the Basket Ball team



C++ if Statement

Syntax:

```
if (condition)
{
    statement;
}
```

```
if (condition)
{
    statement1;
    statement2;
    statement3;
    .
    .
    .
    statementn;
}
```

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
```

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



Example - C++ if Statement

```
ifstatement.cpp
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     float ali_h = 6.2;
7
8     if (ali_h > 6)
9     {
10        cout <<"Ali can become a member of Basket Ball team";
11    }
12
13    return 0;
14 }
```

```
Ali can become a member of Basket Ball team
-----
```

*// Program to print positive number entered by the user
// If the user enters a negative number, it is skipped*

```
#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;
}
```



Operators in if

Relational Operators

Operator	Name	Example
==	Is Equal To	If (x == 10)
!=	Not Equal To	If (x != 10)
>	Greater Than	If (x < 10)
<	Less Than	If (x > 10)
>=	Greater Than or Equal To	If (x >= 10)
<=	Less Than or Equal To	If (x <= 10)

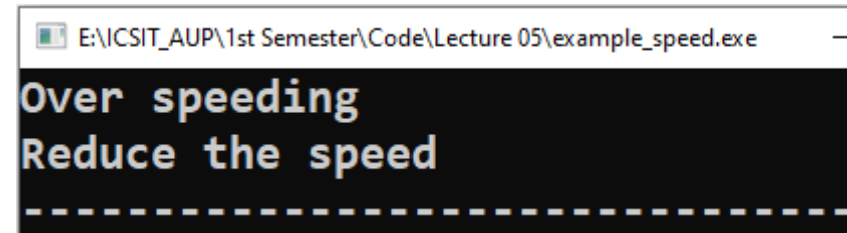
Logical Operators

Operator	Name	Example
&&	Logical AND	If (x == 1 && y == 2)
	Logical OR	If (x == 1 y == 2)
!	Logical Not	If (!x)

Example:

Write a C++ program that tells the user that he is over-speeding and reduce the speed if his speed crosses or equal to 120km/hr.

```
example_speed.cpp
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     int speed = 130;
7
8     if (speed >= 120)
9     {
10        cout << "Over speeding" <<endl;
11        cout << "Reduce the speed";
12    }
13
14    return 0;
15 }
```



```
E:\ICSIT_AUP\1st Semester\Code\Lecture 05\example_speed.exe
Over speeding
Reduce the speed
-----
```

Note:

if is a single selection statement which performs an indicated action only when condition is TRUE; otherwise the action is skipped



If - else Statements

if - else is a double selection statement which performs an action when the condition is TRUE and a different action when the condition is FALSE

The if statement can have an optional else clause. Its syntax is:

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



If - else Statement

The if..else statement evaluates the condition inside the parenthesis

If condition evaluates true,

Code inside body of if is executed

Code inside body of else is skipped from execution

If condition evaluates false,

Code inside body of else is executed

Code inside body of if is skipped from execution

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
```

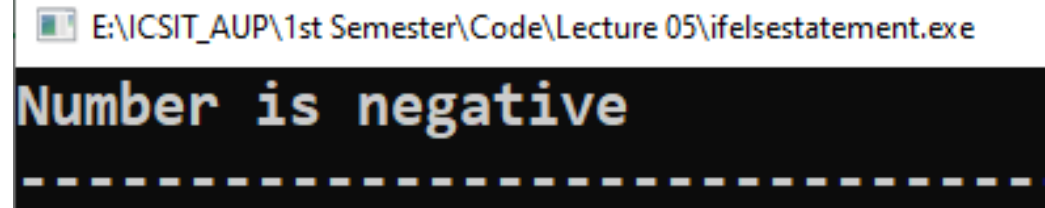


If - else Statement

Example:

Write a C++ program to check whether a number is positive or negative.

```
ifelsestatement.cpp
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int number = -4;
7
8      if (number >= 0)
9      {
10         cout <<"Number is positive";
11     }
12     else
13     {
14         cout <<"Number is negative";
15     }
16
17     return 0;
18 }
```





else if Statement

Used where there are multiple actions to be executed based on different conditions

- if condition is true → action is done
- if condition is false → next condition is checked

Syntax:

If **condition 1** evaluates to true, the code block 1 is executed

If **condition 1** evaluates to false, then condition 2 is evaluated

If **condition 2** is true, the code block 2 is executed

If **condition 2** is false, the code block 3 is executed

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



else if Statement

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
```



else if Statement

Example:

Write a C++ program to check whether a number is positive or negative.

```
elseifstatement.cpp
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int number;
7      cout <<"Enter a number: ";
8      cin >> number;
9
10     if (number > 0)
11     {
12         cout <<"The number you entered is positive.";
13     }
14     else if (number < 0)
15     {
16         cout <<"The number you entered is negative.";
17     }
18     else
19     {
20         cout <<"The number you entered is zero.";
21     }
22
23     return 0;
24 }
```

```
E:\ICSIT_AUP\1st Semester\Code\Lecture 05\elseifstatement.exe
Enter a number: 15
The number you entered is positive.
-----
```

```
E:\ICSIT_AUP\1st Semester\Code\Lecture 05\elseifstatement.exe
Enter a number: -15
The number you entered is negative.
-----
```

```
E:\ICSIT_AUP\1st Semester\Code\Lecture 05\elseifstatement.exe
Enter a number: 0
The number you entered is zero.
-----
```



Nested if Statement

An if statement within the executable block of another if Statement

Conditions are checked in order, i.e. the inner condition is checked only if the outer condition is satisfied

- if condition is true \longrightarrow inner condition is checked
- if inner condition is true \longrightarrow action is done

Syntax:

```
if (condition 1)
{
    // Executes when condition1 is true
    if (condition 2)
    {
        // Executes when condition2 is true
        statement 1;
    }
}
else
{
    statement 1;
    statement 2;
    .
    .
    statement n;
}
```

Nested if Statement

Example:

C++ program to find if number is positive, negative or zero using nested if statements

nestedifstatement.cpp

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int num;
6
7      cout << "Enter a number: ";
8      cin >> num;
9
10     if (num != 0)
11     {
12         if (num > 0)
13         {
14             cout << "The number is positive." << endl;
15         }
16         else
17         {
18             cout << "The number is negative." << endl;
19         }
20     }
21     else
22     {
23         cout << "The number is 0." << endl;
24     }
25     return 0;
26 }
```

E:\ICSIT_AUP\1st Semester\Code\Lecture 05\nestedifstatement.exe

```
Enter a number: 45
The number is positive.
```