Introduction to Computing

Lecture 6

Dr. Naveed Anwar Bhatti

Webpage: naveedanwarbhatti.github.io

C++ Conditions and If Statements

C++ Conditions and If Statements

C++ supports the usual logical conditions from mathematics:

- Less than: `a < b`
- Less than or equal to: `a <= b`
- Greater than: `a > b`
- Greater than or equal to: `a >= b`
- Equal to `a == b`
- Not Equal to: `a != b`

C++ Conditions and If Statements

We can use these conditions to perform different actions for different decisions.

C++ has the following conditional statements:

- Use **`if`** to specify a block of code to be executed, if a specified condition is true
- Use **`else`** to specify a block of code to be executed, if the same condition is false
- Use **`else if`** to specify a new condition to test, if the first condition is false
- Use **`switch`** to specify many alternative blocks of code to be executed

The `if` Statement

Use the **`if`** statement to specify a block of C++ code to be executed if a condition is true.

- Note that **`if`** is in lowercase letters. Uppercase letters (If or IF) will generate an error.

Syntax:

```
if (condition)
{
   // block of code to be executed if the condition is true
}
```

The `if` Statement – Example

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

int main()
{
    if (20 > 18)
    {
        cout << "20 is greater than 18";
    }
    return 0;
}
```

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

int main()
{
    int x = 20;
    int y = 18;
    if (x > y)
    {
        cout << "x is greater than y";
    }
    return 0;
}
```

The `else` Statement

Use the **`else`** statement to specify a block of code to be executed if the condition is **`false`**

- Note that **`else`** is in lowercase letters. Uppercase letters (`Else` or `ELSE`) will generate an error.

Syntax:

```
if (condition)
{
   // block of code to be executed if the condition is true
}
else
{
   // block of code to be executed if the condition is false
}
```

The `else` Statement – Example

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

int main()
{
    int time = 20;
    if (time < 18)
    {
        cout << "Good day.";
    }
    else
    {
        cout << "Good evening.";
    }
    return 0;
}
```

The `else if` Statement

Use the **`else if`** statement to specify a new condition if the first condition is **`false`**

- Note that **`else if`** is in lowercase letters. Uppercase letters (`Else If` or `ELSE IF`) will generate an error.

Syntax:

```
if (condition1)
{
//block of code to be executed if condition1 is true
}
else if (condition2)
{
//block of code to be executed if the condition1 is false and condition2 is true
}
else
{
//block of code to be executed if the condition1 is false and condition2 is false
}
```

The `else if` Statement – Example

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

int main()
{
    int time = 22;
    if (time < 10)
    {
        cout << "Good morning.";
    }
    else if (time < 20)
    {
        cout << "Good day.";
    }
    else
    {
        cout << "Good evening.";
    }
    return 0;
}
```

The **`Switch`** Statement

Use the **`switch`** statement to select one of many code blocks to be executed.

Syntax:

```
switch (expression)
{
    case x:
        // code block
        break;

    case y:
        // code block
        break;

    default:
    // code block
}
```

This is how it works:

- The **`switch`** expression is evaluated once.
- The value of the expression is compared with the values of each **`case`**.
- If there is a match, the associated block of code is executed.
- The **`break`** and **`default`** keywords are optional, and will be described later in this chapter

The `Switch` Statement – Example

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

int main()
{
    int day = 4;
    switch (day) {
    case 1:
        cout << "Monday";
        break;

    case 2:
        cout << "Tuesday";
        break;

    case 3:
        cout << "Wednesday";
        break;

    case 4:
        cout << "Thursday";
        break;

    case 5:
        cout << "Friday";
        break;

    case 6:
        cout << "Saturday";
        break;

    case 7:
        cout << "Sunday";
        break;
    }

return 0;

}
```

- When C++ reaches a break keyword, it breaks out of the switch block.
- A break can save a lot of execution time because it "**ignores**" the execution of all the rest of the code in the switch block.

The `Switch` Statement – Example

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

int main()
{
    int day = 4;
    switch (day) {
    case 1:
        cout << "Monday";
        break;

    case 2:
        cout << "Tuesday";
        break;

    case 3:
        cout << "Wednesday";
        break;
    case 4:
        cout << "Thursday";
        break;
    case 5:
        cout << "Friday";
        break;
    case 6:
        cout << "Saturday";
        break;
    case 7:
        cout << "Sunday";
        break;
    default:
        cout << “Not a valid day";

    }

return 0;

}
```

- The default keyword specifies some code to run if there is no case match
- The default keyword must be used as the last statement in the switch, and it does not need a break.

C++ Exercises

Question 1: Write a program in C++ to check whether the input number is an even or an odd number

Please enter number: <7>
This is an odd number

Question 2: Write a program in C++ to check whether the input alphabet is a vowel or not a vowel. The character is vowel, only if it is equal to a, A, e, E, i, I, o, O, u, U.

Please enter character: <c>
This is not a vowel

C++ Exercises

Solution 1:

```
#include<iostream>
using namespace std;
int main()
{
    int num;
    cout << "Enter a number :";
    cin >> num;

    if (num % 2 == 0)
    {
        cout << "This is an even number";
    }
    else
    {
        cout << "This is an odd number";
    }

}
```

Solution 2:

```
#include<iostream>
using namespace std;
int main()
{
    char ch;
    cout << "Enter an alphabet : ";
    cin >> ch;
    if (ch == 'a' || ch == 'A' || ch == 'e' || ch == 'E' ||
    ch == 'i' || ch == 'I' || ch == 'o' || ch == 'O' ||
    ch == 'u' || ch == 'U')
    {
        cout << "This is a vowel";
    }
    else
    {
        cout << "This is not a vowel";
    }
}
```

Thanks a lot

If you are taking a Nap, **wake up**.....Lecture Over