Unit 3: Conditionals

1. If statement

Suppose Katy says “if it rains, I use my umbrella”.

Similarly, we can have conditional statements in C++.

For example, we can put a condition like:

If x is greater than 5, write ‘x is greater than 5’.

Example 1: Write a program that reads a number from the keyboard and writes ‘x is greater than 5’ if the given number is greater than 5.

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

int main()
{
  int x;
  cin >> x;             // read x from keyboard
  if (x > 5)            // if x is greater than 5
    cout << "x is greater than 5" << endl;
}
```

2. If – else

Suppose Taylor says “if it rains, I use my umbrella else I wear my sunglasses”.

Similarly, in a conditional statement in our program, we can tell the computer what to do if the condition is **false**.

For example,

If x is greater than 5, write ‘x is greater than 5’ else write ‘x is not greater than 5’

Example 2: Write a program that reads a number from the keyboard and writes ‘x is greater than 5’ if the given number is greater than 5. Otherwise, it writes ‘x is not greater than 5’.

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

int main()
{
  int x;
  cin >> x;
  if (x > 5)            // if x is greater than 5
    cout << "x is greater than 5" << endl;
  else                  // otherwise
    cout << "x is not greater than 5" << endl;
}
```

Exercise 1: Read two numbers from the keyboard and find the greater of these two numbers.

Sample input:

```
5
8
```

Sample output:

```
8
```

Question: Can we write the program without using `else` in the `if` statement? How?

REMINDER:

Don’t forget that generally there exist many correct solutions for a certain problem.

3. Relational Operators

We can use the following the following operators that we learned in math in if conditions.

Condition	*Meaning*
`A == B`	A is equal to B
`A != B`	A is not equal to B
`A < B`	A is less than (<) B
`A > B`	A is greater than (>) B
`A <= B`	A is less than or equal to (≤) B
`A >= B`	A is greater than or equal to (≥) B

NOTE:

‘`A = B`’ is different than ‘`A == B`’. ‘`A = B`’ means ‘*assign the value of B to A*’ whereas ‘`A == B`’ means ‘*A is equal to B*’. ‘`A == B`’ has the same meaning of ‘A = B’ in **math**.

Exercise 2: Fill in the following program code so that the program reads two numbers from the keyboard and finds the greater of these two numbers.

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

int main()
{
  int number1;
  int number2;
  cin >> number1;
  cin >> number2;

  if (                        )
    cout << number2 << endl;
  else
    cout << number1 << endl;
}
```

Exercise 3: Read three positive numbers from the keyboard and write the ones with the remainder 3 from the division by 6.

Sample input:

```
11 9 15
```

Sample output:

```
9
15
```

4. Logical Operators

Suppose Dane says “if it rains and I’m outside, I use my umbrella”.

Consider the conditional statement below:

> *If x is greater than 5 and less than 10, write ‘x is less than 4’.*

Example 3: Write a program that reads a number from the keyboard and writes ‘x is less than 4’ if the given number is greater than 5 and less than 10.

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

int main()
{
  int x;
  cin >> x;             // read x from keyboard
  if (x > 5 && x < 10)  // if x is greater than 5 and less than 10
    cout << "x is less than 4" << endl;
}
```

Consider the conditional statement below:

> *If x is greater than 5 or less than 10, write ‘x is less than 4’.*

Suppose Peter says “if it rains or I’m outside, I use my umbrella else I wear my sunglasses”.

Consider the conditional statement below:

> *If x is greater than 5 or less than 10, write ‘x is less than 4’ else write ‘x is greater than 15’.*

Example 4: Write a program that reads a number from the keyboard and writes ‘x is less than 4’ if the given number is greater than 5 or less than 10. Otherwise it writes ‘x is greater than 15’.

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

int main()
{
  int x;
  cin >> x;             // read x from keyboard
  if (x > 5 || x < 10)  // if x is greater than 5 or less than 10
    cout << "x is less than 4" << endl;
  else
    cout << "x is greater than 15" << endl;
}
```

Exercise 4: Write a program that reads a number from the keyboard and writes ‘I liked it’ if the given number is greater than 10 and not equal to 15. Otherwise it writes ‘I didn’t like it’.

Exercise 5: In the following program, to see “x is greater than 15” on the screen what number the user has to enter?

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

int main()
{
  int x;
  cin >> x;
  if (x == 5 || x < 10)
    cout << "x is less than 4" << endl;
  else
    cout << "x is greater than 15" << endl;
}
```

You can also have more than one expression in case of the condition is **true** or **false**.

Example. Suppose that given a number x, we will write ‘small’ and multiply x by 2 if x is less than 10. Otherwise, write ‘large’ on the screen and divide x by 2.

```
if (x < 10)
{
  cout << "small" << endl;
  x = x * 2;
}
else
{
  cout << "large" << endl;
  x = x / 2;
}
```

Here, we write the block of expressions in parenthesis.

General form of `if-else` statement is:

```
if (condition)
{
  (expression)
  (expression)
     :    :
}
else
{
  (expression)
  (expression)
     :    :
}
```

5. If – else if – else

In if statement, we can have a chain of conditions continuing with ‘`else if`’.

Example. Suppose that given a score, we will write ‘not good’, ‘fair’, ‘good’, or ‘awesome’ on the screen with respect to the conditions;

- score < 50,
- 50 ≤ score < 70,
- 70 ≤ score < 90,
- score ≥ 90.

```
  if (score < 50)
    cout << "not good" << endl;
  else if (score >= 50 && score < 70)
    cout << "fair" << endl;
  else if (score >= 70 && score < 90)
    cout << "good" << endl;
  else
    cout << "awesome" << endl;
```