

Tutorial 4

Md. Tanvir Hassan

Email: md.hassan@mail.concordia.ca

The if Statement

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

Example:

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

The else if Statement

```
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  
}
```

Example:

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

Switch Statements

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

```
int day = 4;  
switch (day) {  
    case 6:  
        cout << "Today is Saturday";  
        break;  
    case 7:  
        cout << "Today is Sunday";  
        break;  
    default:  
        cout << "Looking forward to the Weekend";  
}  
// Outputs "Looking forward to the Weekend"
```

Array

- Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.
- To declare an array, define the variable type, specify the name of the array followed by **square brackets** and specify the number of elements it should store:

Example

- `int myNum[3];`
- `int myNum[3] = {10, 20, 30};`
- `string cars[4];`
- `string cars[4] = {"Volvo", "BMW", "Ford", "Mazda"};`

Array

Access the Elements of an Array

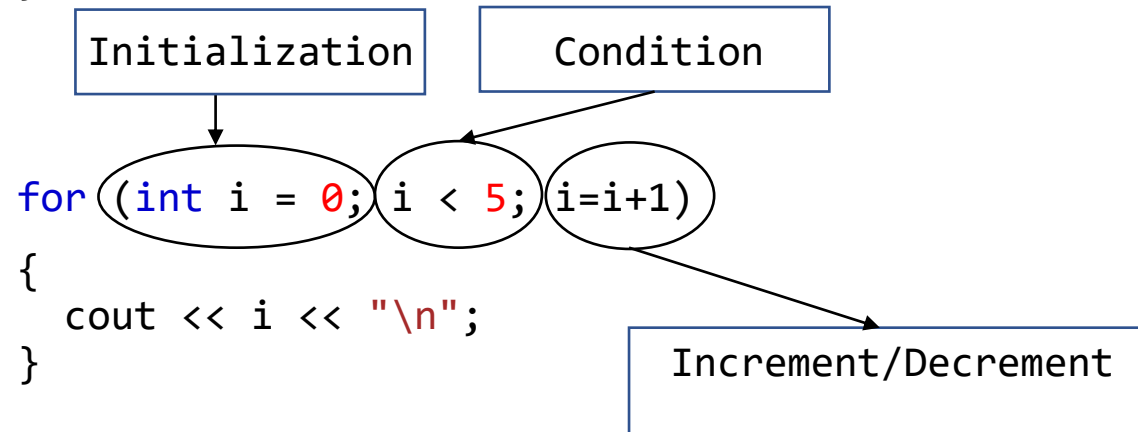
- `string cars[4] = {"Volvo", "BMW", "Ford", "Mazda"};`
- `cout << cars[0];`
- `// Outputs Volvo`

Change an Array Element

- `cars[0] = "Opel";`
- `cout << cars[0];`
- `// Outputs Opel`

For Loop

```
for (statement 1; statement 2; statement 3) {  
    // code block to be executed  
}
```

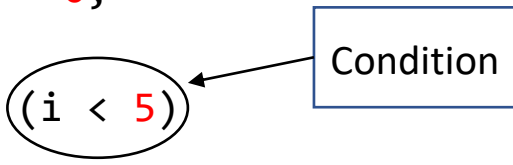


While Loop

```
while (condition) {  
    // code block to be executed  
}
```

```
int i = 0;
```

```
while (i < 5)  
{  
    cout << i << "\n";  
    i++;  
}
```



A diagram illustrating the condition of the while loop. The expression `(i < 5)` is circled in the code. A rectangular box labeled "Condition" has an arrow pointing from it to the circled expression.

Exercises

One interesting application of computers is drawing graphs and bar charts. Write a program that reads five numbers (each between 1 and 30). Assume that the user enters only valid values. For each number that is read, your program should print a line containing that number of adjacent asterisks. For example, if your program reads the number 7, it should print `*****`.

Exercises

(Find the Largest) The process of finding the largest number (i.e., the maximum of a group of numbers) is used frequently in computer applications. For example, a program that determines the winner of a sales contest inputs the number of units sold by each salesperson. The salesperson who sells the most units wins the contest. Write a C++ program that uses a while statement to determine and print the largest number of 10 numbers input by the user. Your program should use three variables, as follows:

- counter: A counter to count to 10 (i.e., to keep track of how many numbers have been input and to determine when all 10 numbers have been processed).
- number: The current number input to the program.
- largest: The largest number found so far.

Exercises

(Printing the Decimal Equivalent of a Binary Number) Input an integer containing only 0s and 1s (i.e., a “binary” integer) and print its decimal equivalent. Use the modulus and division operators to pick off the “binary” number’s digits one at a time from right to left. Much as in the decimal number system, where the rightmost digit has a positional value of 1, the next digit left has a positional value of 10, then 100, then 1000, and so on, in the binary number system the rightmost digit has a positional value of 1, the next digit left has a positional value of 2, then 4, then 8, and so on. Thus the decimal number 234 can be interpreted as $2 * 100 + 3 * 10 + 4 * 1$. The decimal equivalent of binary 1101 is $1 * 1 + 0 * 2 + 1 * 4 + 1 * 8$ or $1 + 0 + 4 + 8$, or 13.

Exercises

(What Prints?) Assume $i = 1$, $j = 2$, $k = 3$ and $m = 2$. What does each statement print?

- a) `cout << (i == 1) << endl;`
- b) `cout << (j == 3) << endl;`
- c) `cout << (i >= 1 && j < 4) << endl;`
- d) `cout << (m <= 99 && k < m) << endl;`
- e) `cout << (j >= i || k == m) << endl;`
- f) `cout << (k + m < j || 3 - j >= k) << endl;`
- g) `cout << (!m) << endl;`
- h) `cout << (!(j - m)) << endl;`
- i) `cout << (!(k > m)) << endl;`