The **while** Loop
- A **loop** repeatedly executes a set of statements until a particular condition is satisfied.
- A **while** loop statement repeatedly executes a target statement as long as a given condition remains **true**.

Syntax:
while (condition) {
statement(s);
}

- The loop iterates while the condition is **true**.
- At the point when the condition becomes **false**, program control is shifted to the line that immediately follows the loop.
- The loop's **body** is the block of statements within curly braces.
- **Example:**

int num = 1;
while (num < 6) {
cout << "Number: " << num << endl;
num = num + 1;
}

/* Outputs
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
*/

- The example above declares a variable equal to 1 (int num = 1).
- The **while** loop checks the condition (num < 6), and executes the statements in its body, which increment the value of **num** by one each time the loop runs.

- After the 5th iteration, **num** becomes 6, and the condition is evaluated to **false**, and the loop stops running.
- The increment value can be changed.
- If changed, the number of times the loop is run will change, as well.

```
int num = 1;
while (num < 6) {
   cout << "Number: " << num << endl;
   num = num + 3;
}

/* Outputs
Number: 1
Number: 4
*/
```

- Without a statement that eventually evaluates the loop condition to **false**, the loop will continue indefinitely.
- The increment or decrement operators are used to change values in the loop.
- **Example:**

```
int num = 1;
while (num < 6) {
   cout << "Number: " << num << endl;
   num++;
}
/* Outputs
Number: 1
Number: 2
Number: 3
Number: 4
Number: 5
*/
```

- num++ is equivalent to num = num + 1.
- A loop can be used to obtain multiple inputs from the user.

- Let's create a program that allows the user to enter a number 5 times, each time storing the input in a variable.

```
int num = 1;
int number;
while (num <= 5) {
    cin >> number;
    num++;
}
```

- The above code asks for user input 5 times, and each time saves the input in the **number** variable.
- Now let's modify our code to calculate the sum of the numbers the user has entered.

```
int num = 1;
int number;
int total = 0;
while (num <= 5) {
    cin >> number;
    total += number;
    num++;
}
cout << total << endl;
```

- The code above adds the number entered by the user to the **total** variable with each loop iteration.
- Once the loop stops executing, the value of **total** is printed.
- This value is the sum of all the numbers the user entered.
- Note that the variable **total** has an initial value of 0.