SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

SPM
"Loops"

By

Dr. Vaibhav P. Vasani

Assistant Professor

Department of Computer Engineering

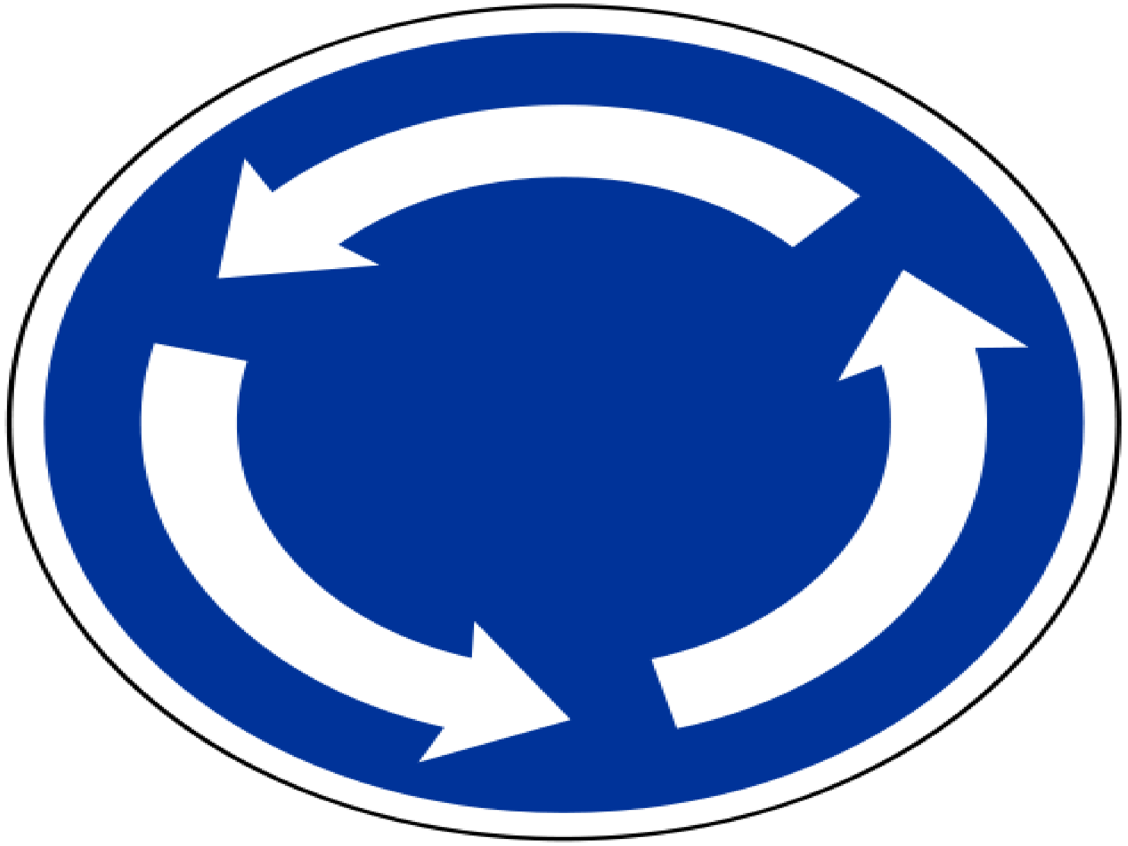
K. J. Somaiya School of Engineering

Somaiya Vidyavihar University

vaibhav.vasani@gmail.com

Somaiya TRUST

Loop analogy (roundabout)



Loop

SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Exiting a Loop

Ninja Cat

Repetition Statements

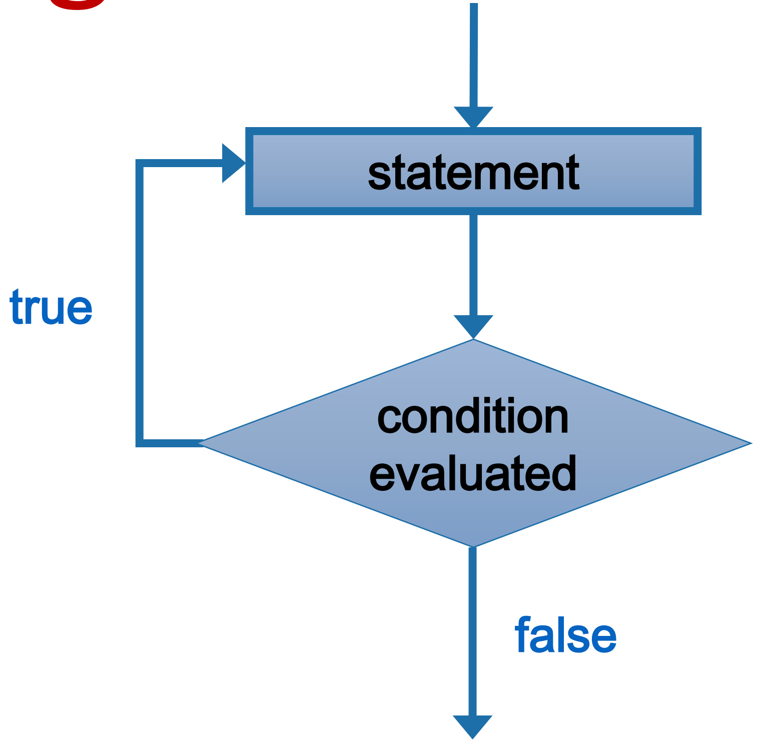
- *Repetition statements* allow us to execute a statement multiple times
- Often they are referred to as *loops*
- C/C++ has three kinds of repetition statements:
  - the *while loop*
  - the *do loop*
  - the *for loop*
- The programmer should choose the right kind of loop for the situation

There are three loop constructs in C++

- do-while loop (or do loop for short)
- while loop
- for loop

Loops = repetition statements

Logic of a do Loop



DO WHILE LOOPS BE LIKE
SHOOT FIRST
ASK QUESTIONS LATER

The do Statement

- A *do statement* has the following syntax:

```
do
{
   statement;
}
while ( condition );
```

- **The `statement` is executed once initially, and then the `condition` is evaluated**
- **The statement is executed repeatedly until the condition becomes false**

SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

The do Statement

- An example of a `do` loop:

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

int main() {
    int count = 0;
    do {
        count++;
        cout << count << endl;
    } while (count < 5);

    return 0;
}
```

- **The body of a `do` loop is executed at least once**

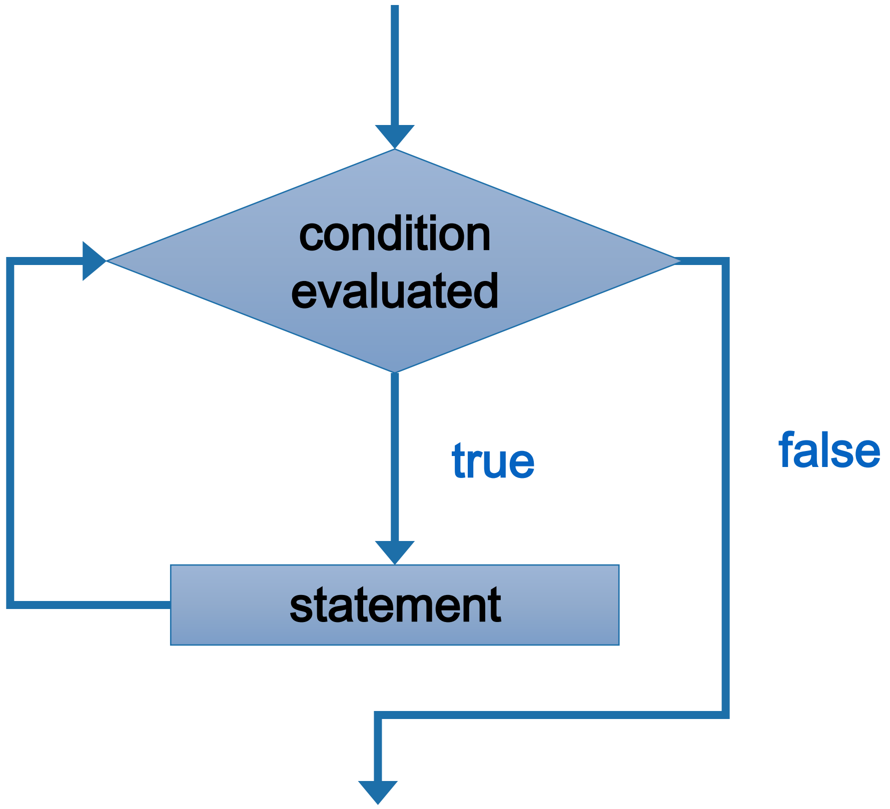
Example: Fixing Bad Keyboard Input

- Write a program that refuses to accept a negative number as an input.

- The program must keep asking the user to enter a value until he/she enters a positive number.

- How can we do this?

Logic of a while Loop



SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

The while Statement

- A *while statement* has the following syntax:

```
while ( condition )
   statement;
```

- **If the `condition` is true, the `statement` is executed**
- **Then the condition is evaluated again, and if it is still true, the statement is executed again**
- **The statement is executed repeatedly until the condition becomes false**

The while Statement

- An example of a while statement:

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

int main() {
    int count = 1;
    while (count <= 5) {
        cout << count << endl;
        count++;
    }
    return 0;
}
```

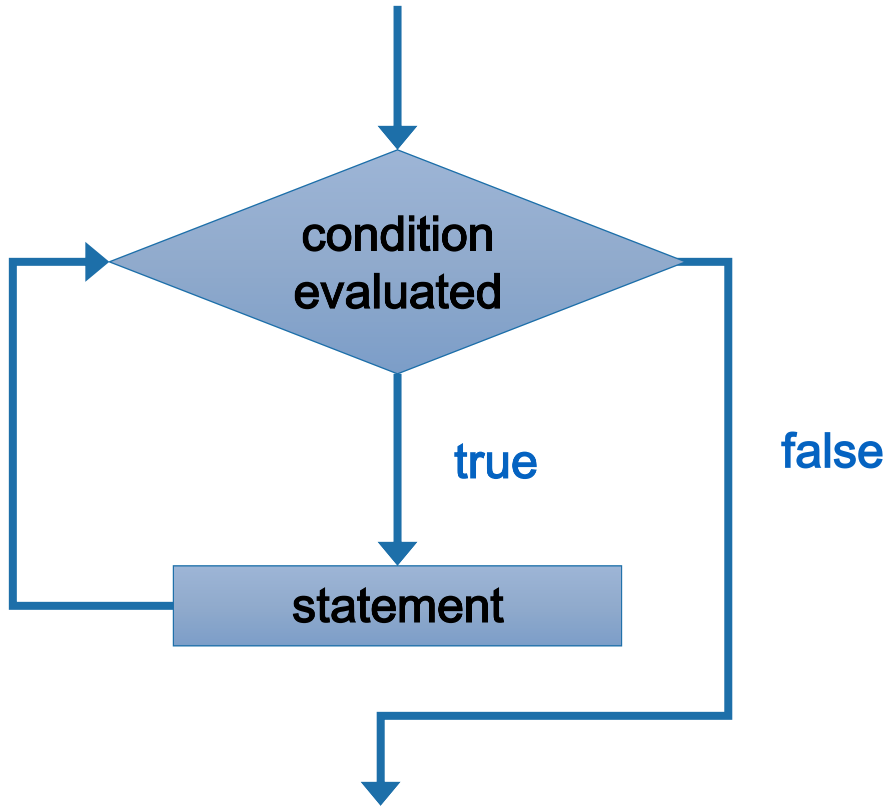
- **If the condition of a `while` loop is false initially, the statement is never executed**
- **Therefore, the body of a `while` loop will execute zero or more times**

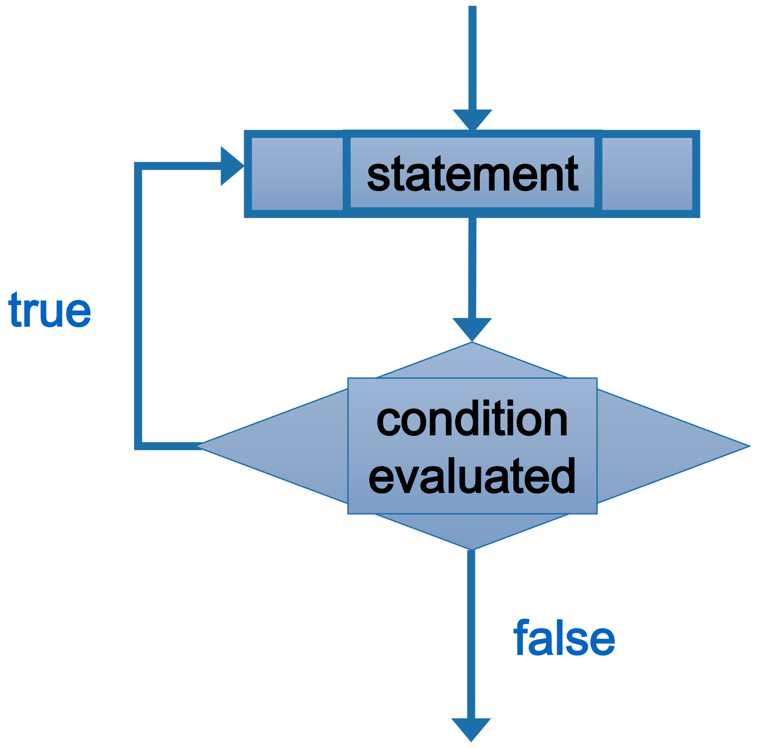
SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

The while Statement

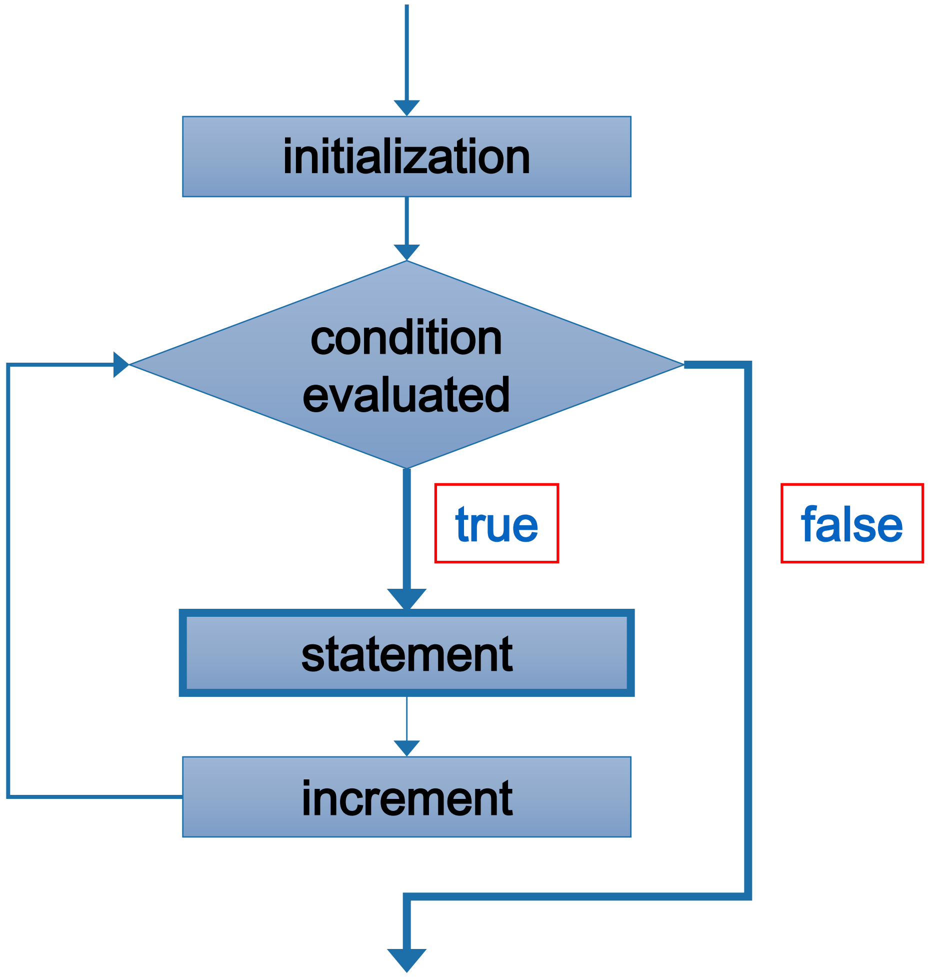
- Let's look at some examples of loop processing
- A loop can be used to maintain a *running sum*
- A *sentinel value* is a special input value that represents the end of input
- A loop can also be used for *input validation*, making a program more *robust*

Comparing while and do





Logic of a for loop



The for Statement

- A *for statement* has the following syntax:

The *initialization* is executed once before the loop begins

The *statement* is executed until the *condition* becomes false

```
for ( initialization ; condition ; increment )
    statement;
```

The *increment* portion is executed at the end of each iteration

The for Statement

- A `for` loop is functionally equivalent to the following `while` loop structure:

```
initialization;
while ( condition )
{
    statement;
    increment;
}
```

The for Statement

- An example of a `for` loop:

```
for (int count=1; count <= 5; count++)
   cout<<count<<endl;
```

- **The initialization section can be used to declare a variable**
- **Like a `while` loop, the condition of a `for` loop is tested prior to executing the loop body**
- **Therefore, the body of a `for` loop will execute zero or more times**

The for Statement

- The increment section can perform any calculation

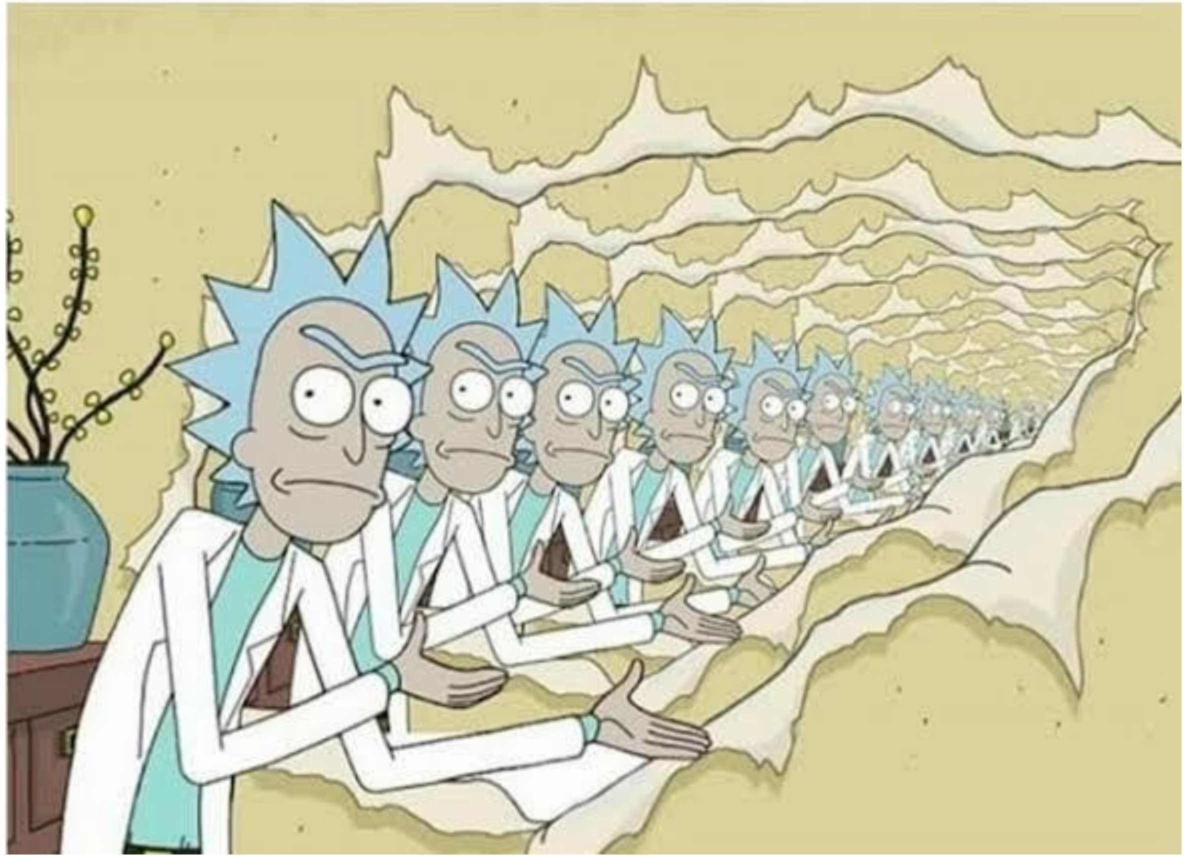
```
Int num;
for (num=100; num > 0; num -= 5)
   cout<<num<<endl;
```

- **A `for` loop is well suited for executing statements a specific number of times that can be calculated or determined in advance**

The for Statement

- Each expression in the header of a `for` loop is optional
- If the initialization is left out, no initialization is performed
- If the condition is left out, it is always considered to be true, and therefore creates an infinite loop
- If the increment is left out, no increment operation is performed

When you forget to break out of the while loop



SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Contact Details

- vaibhav.vasani@somaiya.edu
- vaibhav.vasani@gmail.com

Thank you