Class 07

While Loops to Repeat an Action, For Loops

Repeated Decisions - Loops

```
while (true/false question) {
   statement(s); // executes repeatedly as long as question is true
}
```

If question is true, statement executes repeatedly until the question becomes false

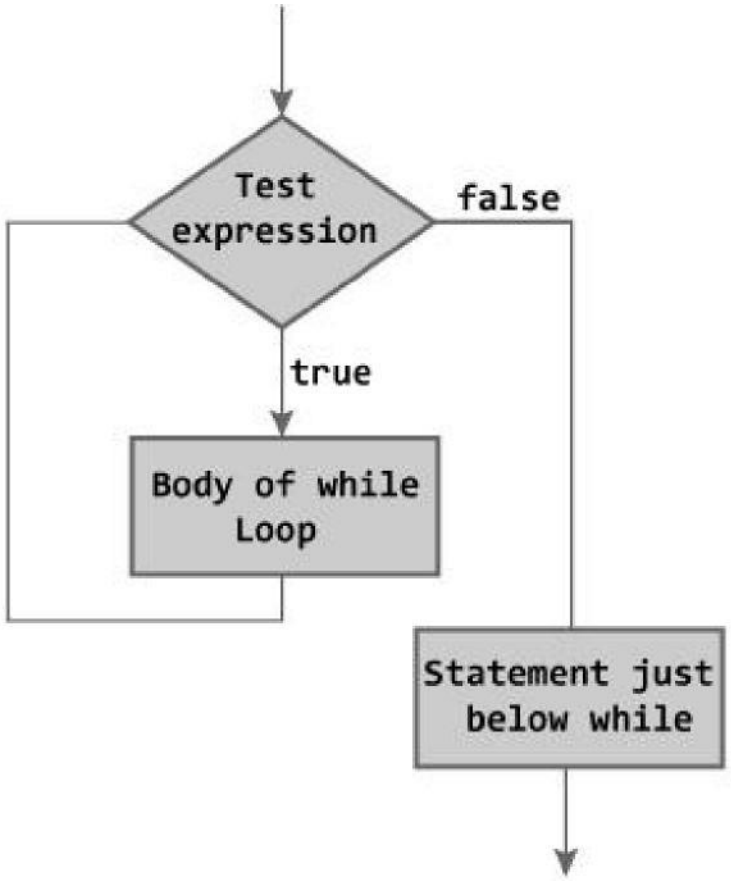

Figure: Flowchart of while Loop

Example 1

- Ask user for positive number. Add up the digits of the number. For example, user enters 1234, computer prints 10 to the monitor.

Example 2

- Ask user for number. Print the first twenty multiples of that number.

While Loop v. For Loop

While Loop	**For Loop**

int counter = 1, num;
cout << “Enter a number: “;
cin >> num;
while (counter <= 20){
   cout << num * counter << “ “;
   counter++;
}
	int num;
	cout << “Enter a number: “;
	cin >> num;
	for (int counter = 1; counter <= 20; counter++){
	   cout << num * counter << “ “;
	}

For Loops

```
for (INITIALIZE COUNTER; IS THE COUNTER VALID?; ADJUST COUNTER){

        ACTION;

}
```

- There are three steps involving the counter, all contained within the for loop parentheses
  - Initialization
  - Check to see if counter is still valid
  - If counter is valid, adjust counter value

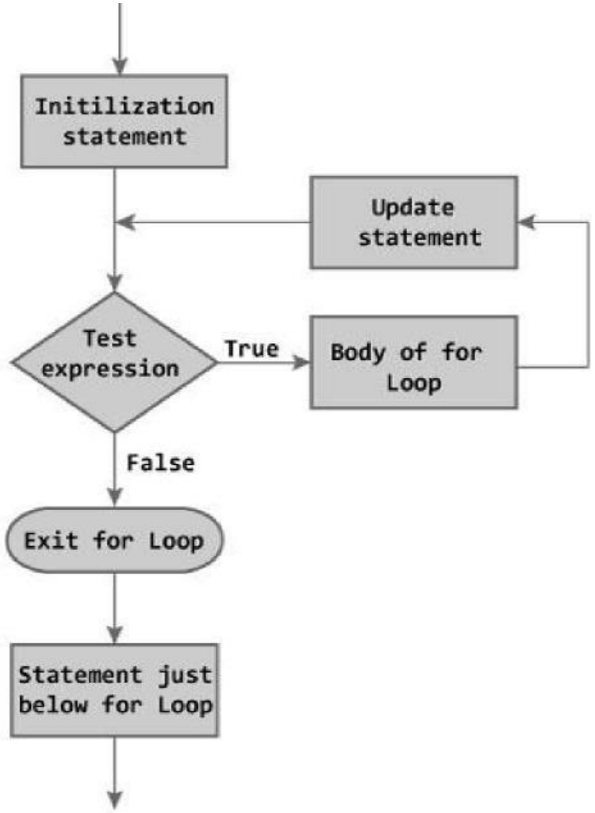

Figure: Flowchart of for Loop

Counters

A counter is an integer variable that is incremented (+1) or decremented (-1) each time an action is performed.

- Count up or count down, depending on the program requirements
- We can use counters to keep track of how many values a user entered
- We can use counters to control how many times a loop runs
- Any time your code has to count something, a counter variable is a tool you can use in your code to do this

Example 3

- Say “hello” ten times.

Example 4

- Print all odd numbers between 1 and 100.

Example 5

- Ask the user for a positive integer. Using a while loop, % 10 and / 10 operations, and a counter, count the number of digits it contains.