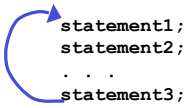
Loops

Section 5.2 - 5.5
Page 255

Loop!

- So far, we can
  - Get input
  - Produce output
  - Calculate
  - Conditionally execute statements
- Loops
  - Perform the same bit of code many times



```
statement1;
statement2;
. . .
statement3;
```

  - Why might we want to do this?

An Example of Repetition

- An example of where we might need to use repetition is if we are calculating the average grade of a class of students
- We would need to continue reading in student grades until we have covered all students
- In pseudocode this might be:

```
While there are more students in the class
   Ask for student grade
```

While Loop

- **while** the **expression** is **true**, loop!

```
while(expression)
{
     statement1;
     statement2;
     . . .
     statement3;
}
statement4;
```

Test the expression

Either

Perform the statements in the loop and return to the `while()` test

or

Move past the loop

13.1 Example: What happens?

```
int number = 0;

while(number < 5)
{
  cout << "Number : " ;
  cout << number << endl;
  cout << "Please enter a number :
";
  cin  >> number;
}
cout << "The final number is: ";
cout << number << endl;
```


13.2 Example: What happens?

```
int number = 0;

while(number < 5 && number != 3)
{
  cout << "Number : " ;
  cout << number << endl;
  cout << "Please enter a number :
";
  cin  >> number;
}
cout << "The final number is: ";
cout << number << endl;
```

13.3 Counters (5.3)

- Counter: A variable that is incremented or decremented each time a loop runs

```
int theCounter = 0;      // initialize the counter
while(theCounter < 2)    // test the counter
{
  cout << "theCounter : " ;
  cout << theCounter << endl;
  theCounter += 1;       // increment the counter
}
```

- What will happen?

13.4 Key Ingredients of `while` loops

- Initialize

  MUST initialize the counter

- Test

  The value of the counter is tested before each iteration

- Update (Increment/Decrement)

  The counter is changed during each loop iteration

If any one of these is missing or incorrect, your loop won't run properly--not at all, too many/few times or infinitely.

13.5 Counters

```
//notice theCounter is now initialized to 1
int theCounter = 1;      // initialize the counter
while( theCounter < 2 ) // test the counter
{
  cout << "theCounter : " ;
  cout << theCounter << endl;
  theCounter += 1;       // increment the counter
}
```

- What will happen?

13.6 Counters

```
int theCounter = 0;      // initialize the counter
while( theCounter < 2 ) // test the counter
{
 theCounter += 1;        // increment the counter
 cout << "theCounter : " ;
 cout << theCounter << endl;
}
```

- What will happen?

13.7 Counters

```
int theCounter = 0;      // initialize the counter
while( theCounter > 2 ) // test the counter
{
 cout << "theCounter : " ;
 cout << theCounter << endl;
 theCounter += 1;        // increment the counter
}
```

- What will happen?

13.8 Practice

- Write a snippet of code that will print all the numbers from 0 to 10000

13.9 Let the user control the Loop (5.4)

- Let the user determine how many times to run the loop

```
int theCounter = 0;            // initialize the counter
int maxValue;

cout << "How many times should we run the loop? ";
cin >> maxValue;

while(                         ) // test the counter
{
 cout << "theCounter : " ;
 cout << theCounter << endl;
                               // increment the counter
}
```


13.10 Practice

- Write a snippet of code that will ask the user for a number. Print the numbers from 0 to the square of the number the user supplied.