

---

# Loops

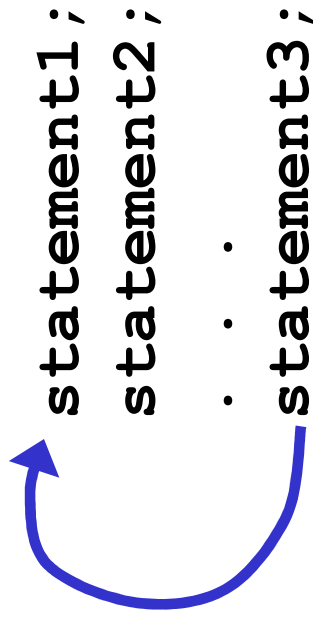
section 5.2, 5.4, 5.7

# 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;
```

A blue curved arrow starts at the end of the line containing 'statement3;' and points back to the beginning of the line containing 'statement1;', illustrating a loop.

- Why might we want to do this?

# While Loop (5.2)

---

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

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

<sup>1</sup>Test the expression

<sup>2</sup> Either

Perform the statements  
in the loop

or

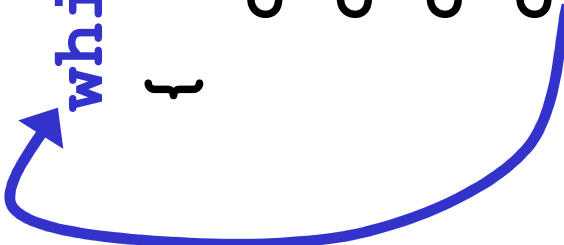
Move past the loop

<sup>3</sup>Repeat

# 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;
```



## Counters (5.4)

---

- 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?

# Key Ingredients of `while` loops

---

- Initialize
- Test
- Update (Increment/Decrement)

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

# Counters

---

```
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?



# 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?

# 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?

# Practice

---

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

# Let the user control the Loop

---

- 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 (
{
    cout << "theCounter : " ;
    cout << theCounter << endl;
}                               // increment the counter
```

# 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.

# Running totals (5.7)

---

- How many hours did you work on assignment 1?

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

// let the user tell us how many times to loop
cout << "How many days did you work on assignment 1? ";
cin >> days;

while (
{
    // increment the counter
}
```

# Practice

---

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

# Practice

---

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



# Exercise

---

- Write a snippet of code that will ask for a student's exam score and then print the appropriate letter grade (A,B,C,D,F).
- Continue asking for exam scores and printing letter grades until the user enters a negative exam score

```
int examScore;
```