
Loops

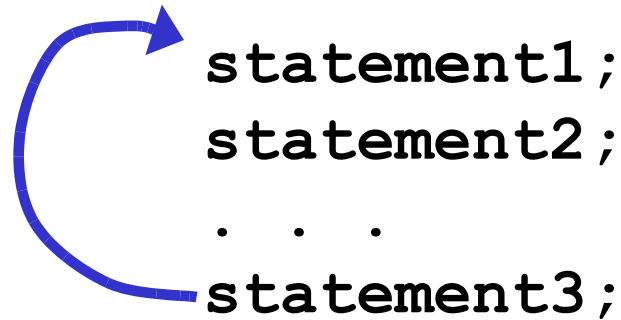
section 5.2, 5.4, 5.7

Loop!

- So far, we can
 - Get input
 - Produce output
 - Calculate
 - Conditionally execute statements

Loops

- Perform the same bit of code many times



- Why might we want to do this?

While Loop (5.2)

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

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

¹Test the expression
² Either
 Perform the statements
 in the loop
 ³Repeat
or
 Move past the loop

Example: What happens?

```
int count = 0;
```

```
while (count < 4)
```

```
{
```

```
    cout << "I'm looping! ";
```

```
    cout << count << endl;
```

```
    count += 1;
```

```
}
```

```
cout << endl << "I'm done looping. ";
```

```
cout << count << endl;
```

Counters (5.4)

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

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

Key Ingredients of `while` loops

- Initialize
- Test
- Update

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

Practice

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

Counters

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

- What will happen?

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

Practice

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

Practice

- Write a piece of code that will ask the user for two integers and display them to the screen. If the sum of the two integers is less than 100, repeat the above action.

Running totals (5.7)

- How many inches of rain did we get last week?

```
const int DAYS_IN_A_WEEK = 7;
int currentDay = 1;           // initialize the counter
double totalRain = 0.0;      // initialize the total
double currentRain;

// loop for each day in a week
while (currentDay <= DAYS_IN_A_WEEK) // test the counter
{
    cout << "How much rain fell on day " << currentDay << ": ";
    cin >> currentRain;
    totalRain += currentRain;

    currentDay ++;           // update the counter
}
cout << "The total rainfall last week was ";
cout << fixed << setprecision(2) << totalRain;
```

Practice

- Write a snippet of code that will ask the user for an integer. Print all the integers from 0 to the number the user supplied as well as the running total of the integers.

```
Enter a number: 3
counter  running  total
      0           0
      1           1
      2           3
      3           6
```

Practice

- Write a snippet of code that will ask the user for a integer. Print the sum of all the *even* integers 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

```
double examScore;
```


Practice

- Write a snippet of code that will print all of the upper case letters, A to Z.

- Write a snippet of code that will print all of the letters, A to Z. Ask the user for the case of the characters.

```
Would you like upper or lower case letters? [U/L]
```