CS150 Intro to CS I

Fall 2014
Chapter 5
Increment, Decrement, Looping, and Files

- Reading: pp. 227-232, 265-284
- Good Problems to Work: p.232 [5.1], p.241 [5.2, 5.3]
Combined Assignments

- We have seen that the same variable can be used on the left hand side of the assignment and on the right hand side:

\[
\begin{align*}
\text{notes} &= \text{notes} / 20; \\
\text{notes} &= \text{notes} \% 20;
\end{align*}
\]

- These are common in programming, so the two operators can be combined as follows:

\[
\begin{align*}
\text{notes} &= \text{notes} /= 20; \\
\text{notes} &= \text{notes} %= 20;
\end{align*}
\]
Combined Assignments

- Combined assignments can be combined with arithmetic operators
  
  ```
  y -= a * 2;
  a /= b + c;
  c %= d - 3;
  ```

- What is the long form of these statements?
Increment and Decrement Operators

- C++ provides a shortcut to increment or decrement a variable by 1
  - Always by 1

```cpp
int x = 99;

x++; // this is equivalent to x += 1
x--; // this is equivalent to x -= 1
```
Prefix and Postfix

Prefix
\[ k = --x; \]
\[ k = ++x; \]
Increment/decrement \( x \) then assign value of \( x \) to \( k \)

Postfix
\[ k = x--; \]
\[ k = x++; \]
Assign value of \( x \) to \( k \), then increment or decrement \( x \)
What is the Output?

```
int y = 0, x = 0, z = 0;
x = y++;
cout << x << " " << y << " " " << z << endl;
y = ++z;
cout << x << " " << y << " " " << z << endl;
z = x++ + 1;
cout << x << " " << y << " " " << z << endl;
```
Tricky ... What is the Output?

```cpp
int count = 0, sum = 0;

while (count++ < 5)
{
    sum += count;
    ++count;
    cout << count << ' ' << sum << endl;
}

cout << count << ' ' << sum << endl;
```
Files

- Data stored in variables is temporary

- We will learn how to write programs that can
  - Create files
  - Write to files
  - Read from files
Steps to Using Files

- There are six steps that must be taken in order to use files in C++
  1. Include proper header files
  2. Define a file stream object
  3. Create a variable to communicate with a file
  4. Open the file
  5. Check that the file opened correctly
  6. Use the file
  7. Close the file
1. Header Files

- To access files you will need
  
  ```
  #include <iostream>
  #include <fstream>
  ```
2. File Stream Objects (Variables)

```cpp
ifstream inputFile;
ofstream outputFile;
fstream inAndOut;
```
3. Opening Files

```cpp
inputFile.open ("filename");
```

- Same syntax for both input and output files
- Filename is a string literal
- Example:
  ```cpp
  ifstream inputFile;
  inputFile.open ("grades.txt");
  ```
4. Check File Opened Correctly

```cpp
inputFile.open ("grades.txt");
if (inputFile.fail())
{
    cout << "Error opening input file ";
    exit (EXIT_FAILURE);
}
```
5. Using File Variables

- Use the input file variable wherever you would use cin
  
  ```cpp
  inputFile >> num;
  ```

- Use output file variable wherever you would use cout
  
  ```cpp
  outputFile << num;
  ```

- Can read/write double, char, int, string
6. Closing Files

- Any files that have been opened must be closed at the end of the program

```cpp
inputFile.close ();
outputFile.close ();
```

Problem:

- A datafile “numbers.txt” exists with one double per line. The final value in the file is -99.0 which is called the sentinel value. The sentinel value is not part of any calculation.

- Write a C++ program that calculates the average of all numbers in the file “numbers.txt”