**Lab Experience Seven**

**Lab Exercises**

1. Convert the following while loops into for loops:

a)

int x = 1;

while(x > 0){

cout << “Enter a number “;

cin >> x;

}// end while

**int x;**

**for(x = 1; x > 0;) {**

**cout << “Enter a number “;**

**cin >> x;**

**}**

b)

int i = 0;

while ( i < 50){

cout << “i is “ << i << endl;

i++;

}// end while

**for(int i = 0; i < 50; i++) {**

**cout << “i is “ << i << endl;**

**}**

**Programming Exercise 1:**

Write a C++ program that reads a file and counts the number of alphabetic characters, number of uppercase letters, number of punctuation characters, and the number of whitespace characters in the file. Utilize the get function to process the file one character at a time. The data file to use is the Gettysburg Address, which is found below. The program should produce a report containing the above information. Download the file Gettysburg.txt from D2L. The text of the speech is shown below.

Copy and paste your program into your word document. Capture the output window and paste it below your program.

Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battle-field of that war. We have come to dedicate a portion of that field, as a final resting place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we cannot dedicate -- we cannot consecrate -- we cannot hallow -- this ground. The brave men, living and dead, who struggled here, have consecrated it, far above our poor power to add or detract. The world will little note, nor long remember what we say here, but it can never forget what they did here. It is for us the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task remaining before us -- that from these honored dead we take increased devotion to that cause for which they gave the last full measure of devotion -- that we here highly resolve that these dead shall not have died in vain -- that this nation, under God, shall have a new birth of freedom -- and that government of the people, by the people, for the people, shall not perish from the earth.

/\*

\* Programmer: Johnathan Lee

\* Class: CSCI 1106

\* Lab#: 7

\*

\* Description: Counts the number of alphabetic characters, number of uppercase letters, number of punctuation characters, and the number of whitespace characters in the file.

\*/

#include <ctime>

#include <cstdlib>

#include <iostream>

#include <fstream>

using namespace std;

int main() {

char curChar, lastChar;

int numAlpha = 0, numUpper = 0, numPunctuation = 0, numWhitespace = 0;

ifstream inFile;

inFile.open("gettysburg.txt");

if (!inFile) {

cout << "Error opening file. Terminating program early.\n\n";

exit(1106);

}

while (!inFile.eof() && !inFile.bad()) {

curChar = inFile.get();

if (curChar >= 'A' && curChar <= 'Z') {

numAlpha++;

numUpper++;

} else if (curChar >= 'a' && curChar <= 'z') {

numAlpha++;

} else if (curChar >= '!' && curChar <= '/' && lastChar != '-') {

numPunctuation++;

} else if (curChar == ' ' || curChar == 9 /\*tab\*/ || curChar == '\n') {

numWhitespace++;

}

lastChar = curChar;

}

cout << "Alphabetic: " << numAlpha << endl

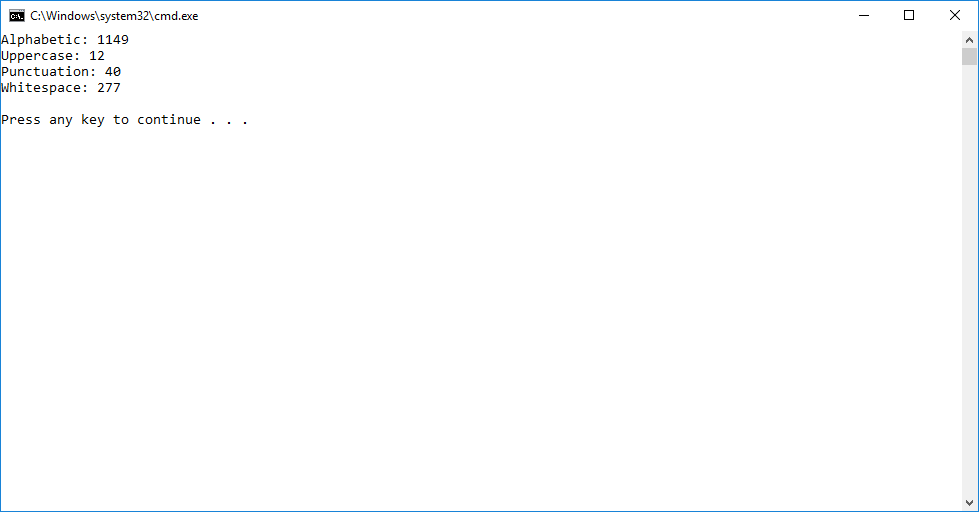
<< "Uppercase: " << numUpper << endl

<< "Punctuation: " << numPunctuation << endl

<< "Whitespace: " << numWhitespace << "\n\n";

return 0;

}



**Programming Exercise 2**

Mighty Casey plays centerfield for the Toledo Mudhens and has the following lifetime hitting percentages:

|  |  |
| --- | --- |
| Out | 35% |
| Walk | 16 % |
| Single | 20% |
| Double | 15% |
| Triple | 9% |
| Home Run | 5% |

The above is called a probability distribution. A probability distribution is used in predicting the likelihood of an event occurring. See the following explanation of probability distributions using the example of flipping a coin at: <http://stattrek.com/Lesson2/ProbabilityDistribution.aspx>. Notice the distribution adds up to 100%.

Write a program to simulate a large number of times at bat (1000 or more) for Mighty Casey counting the number of outs, walks, singles, etc. to predict Mighty Casey’s batting average for next season and slugging percentage. This means you have to generate a random number (hint: between 1 and 100) and based upon the value (i.e. probability) will determine if Mighty Casey gets a hit or an out. If it is a hit, then your program needs to determine what type of hit was based upon the probabilities given in the table above.

Use the formulas below to calculate the batting average and slugging percentage of Mighty Casey.

batting average = number of hits / (number of times at bat – number of walks)

slugging percentage = (number of singles + number of doubles \* 2 + number of triples \* 3

+ number of homeruns \* 4) /(number of times at bat – number of walks)

Copy and paste your program into your word document. Capture the output window and paste it below your program.

/\*

\* Programmer: Johnathan Lee

\* Class: CSCI 1106

\* Lab#: 7

\*

\* Description: Calculates batting average and slugging percentage based on a simulation.

\*/

#include <ctime>

#include <cstdlib>

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

// Probability ranges for all possible actions.

const int OUT\_MAX = 35, SINGLE\_MAX = 20 + OUT\_MAX, WALK\_MAX = 16 + SINGLE\_MAX,

DOUBLE\_MAX = 15 + WALK\_MAX, TRIPLE\_MAX = 9 + DOUBLE\_MAX,

HOMERUN\_MAX = 5 + TRIPLE\_MAX; // This should be 100. Can be used to double check things.

// How many times will we run the sim? (The assignment said 1000 or more, so I

// assumed this was a choice and left it const.)

const int NUM\_SIMS = 1000;

int main() {

int curRand, // A temp variable for each random number.

numHits,

numOuts = 0,

numWalks = 0, // Number of each time

numSingles = 0,

numDoubles = 0,

numTriples = 0,

numHomes = 0;

double battingAvg, sluggingPct;

srand(time(NULL));

for (int i = 0; i < NUM\_SIMS; i++) {

curRand = 1 + rand() % 100; // Rand in [1,100]

if (curRand <= OUT\_MAX)

numOuts++;

else if (curRand <= SINGLE\_MAX)

numSingles++;

else if (curRand <= WALK\_MAX)

numWalks++;

else if (curRand <= DOUBLE\_MAX)

numDoubles++;

else if (curRand <= TRIPLE\_MAX)

numTriples++;

else // <= 100

numHomes++;

}

numHits = NUM\_SIMS - numOuts;

battingAvg = static\_cast<double>(numHits) / (NUM\_SIMS - numWalks);

sluggingPct = (numSingles + numDoubles \* 2.0 + numTriples \* 3.0 + numHomes \* 4.0)

/ (NUM\_SIMS - numWalks);

cout << fixed << showpoint << setprecision(2);

cout<< "Number of hits: " << numHits << endl

<< "Number of outs: " << numOuts << endl

<< "Number of walks: " << numWalks << endl

<< "Number of singles: " << numSingles << endl

<< "Number of doubles: " << numDoubles << endl

<< "Number of triples: " << numTriples << endl

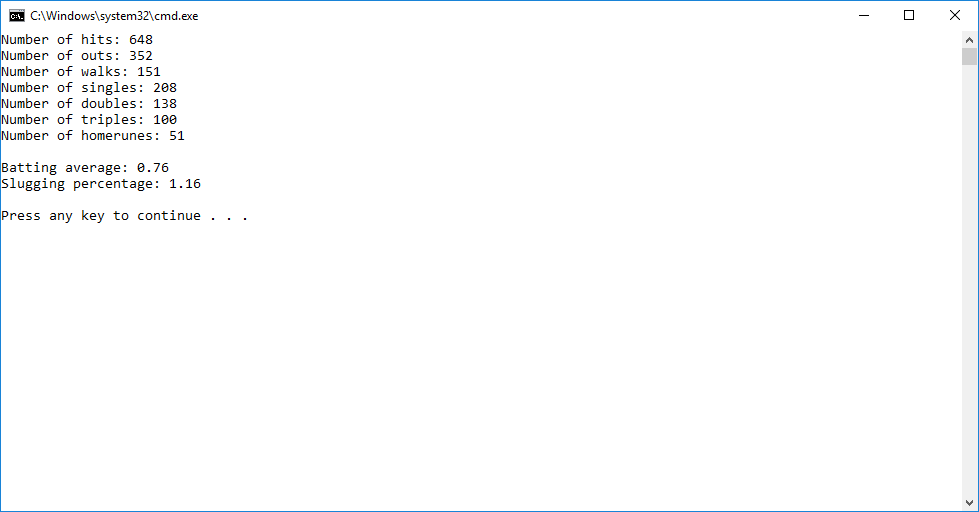
<< "Number of homerunes: " << numHomes << endl << endl

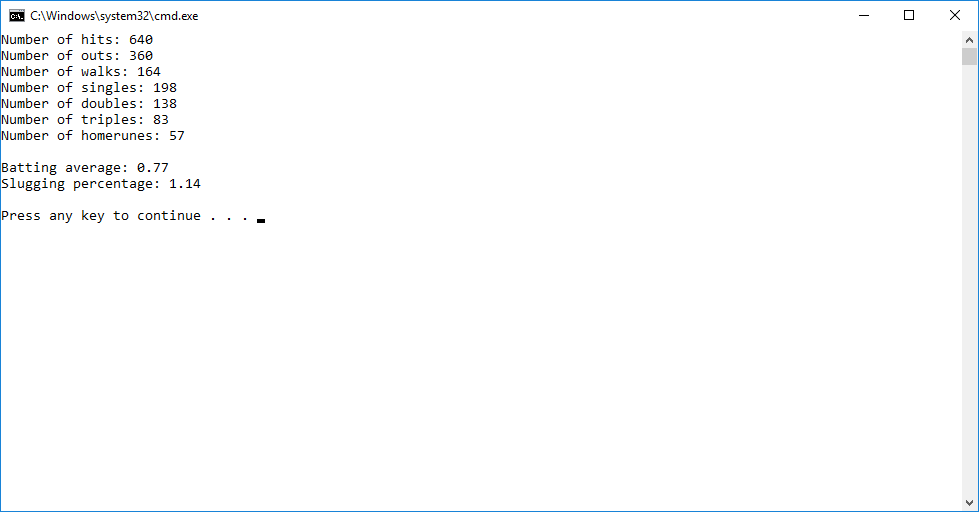
<< "Batting average: " << battingAvg << endl

<< "Slugging percentage: " << sluggingPct << endl << endl;

return 0;

}

.



**Due Dates:**

As specified on the D2L assignment folder for lab 7.

**What to hand in:**

1. Hand in a print out of your word document.
2. Hand in a print out of your program.
3. Compress the .cpp and the word processed document into a single compressed file called **{yourname}Lab7.zip** e.g. timwrennlab8.zip Note:**If your name is not part of the zip filename, I will not open the zipped file.** A popular file compression program is [winzip](http://www.winzip.com/), which you can download to use on a trial basis.
4. Place the compressed file into the lab experience seven assignment folder.