**Andrew Tran**

**Lab Experience Nine**

**Objectives:**

1. Utilizing file stream objects.
2. Understanding the for loop.

**Background**

The previous lab introduced you to the concept of the repetition structure and how it is implemented in C++. This lab will present the **for** loop construct and how it is common to nest loops within each other to accomplish the task presented. It is a common process to solve problems by breaking the problem up into smaller tasks that can be easily solved. This process in Computer Science is called **modularity** and is implemented in C++ with functions. A function is a segment of code that solves one task and one task only. For example the function **pow** only performs exponential operations and nothing else. In addition, it is the programmer’s responsibility to provide the correct information to the **pow** function, otherwise unpredictable results might occur, i.e. **pow(0,0)** should not work and either an error message will be displayed or a garbage value will be returned.

**The for Loop**

Recall the syntax of the while loop:

**while(expression){**

**statement-1;**

Recall if more than one statement follows the while, it must be enclosed between {}.

**statement-2;**

**statement-3;**

**// continuing**

**statement-n;**

**}**

The syntax of the **for** loop is as follows:

|  |  |
| --- | --- |
| If only one statement following the **for** | More than one statement following the **for** |
| **for(initialization; test expression; update expression)**  **statement;** | **for(initialization; test expression; update expression){**  **statement-1;**  **statement-2;**  **// more statements**  **statement-n;**  **}** |

The three components of the **for** loop are as follows:

1. Initialization expression is used to set an initial value of a variable that will be used in the test expression.
2. The test expression is an expression similar to what has been used in the while, if-else, etc. If the test is true the loop will execute, if false the loop will not execute.
3. The update expression updates the variable used in the test expression to determine if the loop will continue or not.

As an example consider the following code segment from the previous lab and how it has been rewritten using a **for** loop:

|  |  |
| --- | --- |
| **int sum = 0;**  **int count = 1; // initialization**  **while (count <= 5){ // test**  **sum = sum + count;**  **count++; // update**  **}// end while** | int sum = 0;  for(int count = 1; count <=5; count++)  sum += count; |

Notice the similarities between the two constructs.

1. Initialization is the same and is only executed once.
2. The test expression is evaluated and if true the loop is executed, if false the loop is not executed.
3. The update of the variable controlling the loop is executed after the loop body has been executed. That is, it is the last thing that happens.
4. Repeat steps 2 and 3 until the test expression is false.

**Nested Loops**

It is common to place loops inside of other loops. The inner loop will always complete its task before the outer loop and it is common the outer loop will execute again. An example of this is multiplication tables. Write a C++ program to produce the first 3 rows of a multiplication table.

The code segment to accomplish this can be written either using a for loop or a while loop.

|  |  |
| --- | --- |
| while loops | for loops |
| int i = 1;  while(i <= 3){  int j = 1;  while(j <=3){  product = i \* j;  cout << setw(3) << product;  j++; // update  }// end inner loop  cout << endl; // go to the next line.  i++; // update  }// end outer loop | for(int i = 1; i <= 3; i++){  for(int j = 1; j <= 3; j++){  product = i \* j;  cout << setw(3) << product;  }  cout << endl; // go to the next line  } |

The output of the above code segments is the same and will be:

|  |  |
| --- | --- |
| **Output** | **A partial trace of the above program segment**  **The value of i remains fixed until the inner loop completes. Also note the value of j is reset to the value of one after i is incremented. A duplicate definition error does not occur because of block scope.** |
| 1 2 3  2 4 6  3 6 9 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **i** | **i <= 3** | **j** | **j <=3** | **product** | | 1 | True | 1 | True | 1 | | 1 |  | 2 | True | 2 | | 1 |  | 3 | True | 3 | | 1 |  | 4 | False | 3 | | 2 | True | 1 | True | 2 | | 2 | True | 2 | True | 4 | | 2 |  | 3 | True | 6 | |

**Lab Exercises**

**Directions:**

Start Microsoft word and record the questions and answers to all of the exercises in the lab word document   
Answer the following questions based on material presented in lecture and found in chapters 1-5 of the textbook.

**Short Answer**

1. Perform a hand trace to determine the output of the following program segments. A snapshot of the program execution will receive no credit.

a)

**cout << "12345678901234567890" << endl;**

**for(int i = 3; i >=0; i--)**

**cout << setw(3) << i ;**

123

123

123

123

b)

**for(int i = 1; i < 5; i+= 2){**

**cout << “i = “ << i << endl;**

**for(int j = 5; j > 0; j-= 2){**

**value = i \* j;**

**cout << “ value = “ << value << endl;**

**cout << “ j = “ << j << endl;**

**}**

**cout << endl;**

**}**

i=1

value=5

j=5

i=3

value=9

j=3

2. Convert the following while loops into for loops:

a)

**int x = 1;**

**while(x > 0){**

**cout << “Enter a number “;**

**cin >> x;**

**}// end while**

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(i=0;i<50;i++){

cout << “i is “ << i << endl;

}

**Exercise 1**

You are interested in analyzing the frequency of numbers selected for the Powerball lottery. You have downloaded the previous year’s Powerball numbers which are saved in a text file called **pb\_2014.txt** and each record in the file has the following format:

01/01/14 15 24 40 48 52 23 x PP

01/04/14 19 20 37 41 58 14 x PP

01/08/14 10 28 39 47 58 22 x PP

01/11/14 10 15 33 48 54 34 x PP

Unfortunately this is an incompatible file format for the program you are using to analyze the frequency. The correct record file format needed by the program is shown below:

15 24 40 48 52 23

19 20 37 41 58 14

10 28 39 47 58 22

10 15 33 48 54 34

Write a program to convert the **pb\_2014.txt** file into the required file format shown above saving the new format to the file **powerball.txt**.

Copy and paste your program into your word document. Copy and paste the first 10 records of each file directly below your program.

#include <iostream>

#include <fstream>

#include <iomanip>

using namespace std;

int main()

{

ifstream myfile;

myfile.open("pb\_2014.txt");

int countone;

int counttwo;

double number;

ofstream writingfile;

writingfile.open("powerball.txt");

for (countone = 0; countone < 100; countone++)

{

for (counttwo = 0; counttwo < 6; counttwo++)

{

myfile.ignore(10, ' ');

myfile >> number;

writingfile << setw(2) << setfill('0') << number << ' ';

}

writingfile << endl;

myfile.ignore(10);

}

myfile.close();

writingfile.close();

cout << "Writing to powerball.txt ---- Done." << endl;

return 0;

}

Pb\_2014:

01/01/14 15 24 40 48 52 23 x PP

01/04/14 19 20 37 41 58 14 x PP

01/08/14 10 28 39 47 58 22 x PP

01/11/14 10 15 33 48 54 34 x PP

01/15/14 07 08 09 24 29 25 x PP

01/18/14 13 14 19 31 38 25 x PP

01/22/14 01 02 07 09 55 29 x 03

01/25/14 08 12 18 55 57 02 x 02

01/29/14 11 23 28 32 47 20 x 02

02/01/14 05 12 15 27 38 07 x 02

Powerball.txt:

15 24 40 48 52 23

19 20 37 41 58 14

10 28 39 47 58 22

10 15 33 48 54 34

07 08 09 24 29 25

13 14 19 31 38 25

01 02 07 09 55 29

08 12 18 55 57 02

11 23 28 32 47 20

05 12 15 27 38 07

**Due Dates:** According to the due date posted for the drop box folder.

**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}Lab9.zip** e.g. timwrennlab9.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 9 dropbox folder.