**ITCS 1212L**

**Post-Lab 7**

1. (9 points) You were told that when you are using the for-loop, you should determine the initial value of the loop control variable, the condition and the update depending on the logic of your program. What will happen if:
   1. You forget to initialize your loop control variable. This depends on if you leave that section blank and include the semi-colon, or if you completely forget the loop control variable AND the semi-colon. If you forget the semi-colon, the program will not recognize your for-loop parameters and it will not compile. However, if you simply leave the initialization blank and leave in a semi-colon, the variable in the condition will have whatever value the program had previously assigned it, OR a random number determined at the start of the program (assuming it was declared and not assigned a value).
   2. You forget to include your condition. If left blank without its semi-colon, as in part 1.a, the program will not compile. However, if the semi-colon is present without a conditional statement, the program will ready it as “loop when this condition is true. Also, this statement is always true.” That means your program will run, but it will run in an infinite loop, continually incrementing or decrementing your control variable.
   3. You forget to include the update rule. The update rule isn’t followed by a semi-colon, so the program will compile no matter how you forget the update rule. The problem is, if you do not have a way to increment or decrement your variable INSIDE the loop, the control variable will never change its value and the program will be stuck in an infinite loop.
2. (12 points) Use De’ Morgan’s law to convert each of these expressions to a form without parenthesis: (Look up De’ Morgan’s Law online)
   1. ! (A || B)

Not A AND Not B

!A && !B

* 1. ! (A && B)

Not A OR Not B

!A || !B

* 1. ! (A && B && C)

Not A OR Not B OR Not C

!A || !B || !C

* 1. ! (A || !B)

Not A AND Is B

!A && B

1. (9 points) What are the 3 types of loops available in C++? Determine the ones that are pre-test loops as well as post-test loops.

Pre-test: For-Loop, While Loop

Post-test: Do-While Loop

1. (5 points) Which loop is guaranteed to execute at least one time ?

A Do-While Loop will execute at least one time because it is a post-test loop.

1. (5 points) How many times will the following loop run?

char x;

do {

cout << “Enter your input. Correct choices are A, B, and C: “ << endl;

cin >> x;

switch ( x )

{

case ‘A’:

….

case ‘B’:

….

case ‘C’:

….

}

} while(x != ‘A’ || x != ‘B’ || x != ‘C’);

Infinitely. There is no condition set that will exit this loop without further modification.

1. Create a multiplication table using a nested for-loop that the outer loop goes from 1 to 10 and the inner loop goes from 1 to 10 as well. Inside the inner loop you display the result of the multiplication of the two loop counters. (20 points)

I probably used a lot more code than was necessary for this project, but it made the output very appealing. I’m going to post my algorithm here to show my understanding of the task, because the code itself is really long:

for (x = 1; x <= 10; x++)

{

for (y = 1; y <= 10; y++)

{

product = x \* y;

cout << product;

}

}

------------------------------------------------------------------------------------------

[Here’s my full code…]

**#include <iostream>**

**#include <cstdlib>**

**#include <time.h>**

**using namespace std;**

**void xLoop(int);**

**void yLoop(int, int);**

**int main()**

**{**

**int x(0), y(0);**

**cout << endl << "\t\t Welcome to my Multiplication Table!" << endl;**

**cout << "\t\t---------------------------------------" << endl << endl;**

**cout << "|\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\n";**

**cout << "| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |\n";**

**for (x=1; x <= 10; x++)**

**{**

**xLoop(x);**

**for (y=1; y <= 10; y++)**

**{**

**yLoop(x, y);**

**}**

**}**

**cout << "|\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\n";**

**cout << endl; //For good measure :P**

**return 0;**

**}**

**void xLoop(int x)**

**{**

**cout << "|\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\*\*\*\*\*\*|\n";**

**if (x == 1)**

**{**

**cout << "| 1 | ";**

**}**

**else if (x == 2)**

**{**

**cout << "| 2 | ";**

**}**

**else if (x == 3)**

**{**

**cout << "| 3 | ";**

**}**

**else if (x == 4)**

**{**

**cout << "| 4 | ";**

**}**

**else if (x == 5)**

**{**

**cout << "| 5 | ";**

**}**

**else if (x == 6)**

**{**

**cout << "| 6 | ";**

**}**

**else if (x == 7)**

**{**

**cout << "| 7 | ";**

**}**

**else if (x == 8)**

**{**

**cout << "| 8 | ";**

**}**

**else if (x == 9)**

**{**

**cout << "| 9 | ";**

**}**

**else**

**{**

**cout << "| 10 | ";**

**}**

**}**

**void yLoop(int x, int y)**

**{**

**int product(0);**

**product = x \* y;**

**if (product < 10 && y != 10)**

**{**

**cout << product << " | ";**

**}**

**else if (product < 100 && y != 10)**

**{**

**cout << product << " | ";**

**}**

**else if (product < 100 && y == 10)**

**{**

**cout << product << " |";**

**}**

**else**

**{**

**cout << product << " |";**

**}**

**if (y == 10)**

**{**

**cout << endl;**

**}**

**}**

1. Write a program that reads several integers from a file called inputFile. The program shows the sum and average of all these integers. Write the program in two different ways:
   1. Assume that there are exactly 10 data in the files on 10 different rows. Simulate it and make sure the program works. (15 points)

**[Keep in mind that for this code to work on a machine other than mine, the directory will have to be changed during the inutFile.open argument.]**

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int main()**

**{**

**ifstream inputFile;**

**int x(0), number(0), total(0);**

**float average(0);**

**inputFile.open("C:\\Users\\Goat Prime.Goat\\Desktop\\inputFile.txt");**

**if (inputFile)**

**{**

**for (x = 1; x <= 10; x++)**

**{**

**inputFile >> number;**

**total += number;**

**}**

**inputFile.close();**

**average = total / 10;**

**cout << endl << "The Total is: " << total << endl;**

**cout << "The Average is: " << average << endl;**

**}**

**else**

**{**

**cout << endl << "Error. Could not read file!" << endl;**

**}**

**return 0;**

**}**

* 1. Assume that you do not know the number of data in files and all you know is the last number is -999. (15 points)

[Again, the file name will have to be changed to run this on another device.]

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int main()**

**{**

**ifstream inputFile;**

**int x(0), number(0), total(0);**

**float average(0);**

**inputFile.open("C:\\Users\\Goat Prime.Goat\\Desktop\\inputFile.txt");**

**if (inputFile)**

**{**

**while (inputFile >> number)**

**{**

**if (number != -999)**

**{**

**total += number;**

**x++;**

**}**

**else**

**{**

**average = total / x;**

**cout << endl << "The Total is: " << total << endl;**

**cout << "The Average is: " << average << endl;**

**}**

**}**

**}**

**else**

**{**

**cout << endl << "Error. Could not read file!" << endl;**

**}**

**return 0;**

**}**