**ITCS 1212L**

**PostLab 8**

**Loops and Functions**

1. **Pattern Displays (20 points)**

Write a program that uses a loop to display Pattern A below, followed by another loop that displays Pattern B.

|  |  |
| --- | --- |
| Pattern A | Pattern B |
| +  ++  +++  ++++  +++++  ++++++  +++++++  ++++++++  +++++++++  ++++++++++ | **++++++++++**  **+++++++++**  **++++++++**  **+++++++**  **++++++**  **+++++**  **++++**  **+++**  **++**  **+** |

Pattern A

#include <iostream>

using namespace std;

int main()

{

cout << endl << endl;

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

{

for (int y = 1; y <= x; y++)

{

cout << "+";

}

cout << endl;

}

return 0;

}

Pattern B

#include <iostream>

using namespace std;

int main()

{

cout << endl << endl;

for (int x = 10; x >= 1; x--) //This is what I had to change from pattern A

{

for (int y = 1; y <= x; y++)

{

cout << "+";

}

cout << endl;

}

return 0;

}

1. **Using Files- Numeric Processing (20 points)**

If you have downloaded this book’s source code from companion Web site, you will find a file named Random.txt in the Chapter 05 folder. (The companion site is [www.pearsonhighered.com/gaddis](http://www.pearsonhighered.com/gaddis).) This file contains a long list of random numbers. Copy the file to your hard drive and then write a program that opens the file, reads all the numbers from the file, and calculates the following.=

1. The total count of numbers in the file
2. The sum of all the numbers in the file (a running total)
3. The average of all the numbers in the file

The program should display A, B, and C.

For this task, I could not find this list of numbers, so I created my own notepad file and populated it with my own random numbers. The code I wrote for this should work for the task regardless of which file is used, so it should still be good to go if running with the companion site file (assuming the directory name is changed accordingly within my code). [My file was called “random.txt”]

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

ifstream randomFile;

int x(0);

float average(0), total(0), number(0);

cout << endl;

randomFile.open("C:\\Users\\Goat Prime.Goat\\Desktop\\random.txt");

if (randomFile)

{

while (randomFile)

{

randomFile >> number;

x++;

total += number;

}

randomFile.close();

average = total / (x-1);

cout << "Your file contained " << x-1 << " numbers." << endl;

cout << "Your file's running total is: " << total << endl;

cout << "The average for your file is: " << average << endl;

}

else

{

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

}

return 0;

}

1. **Random Number Guessing Game Enhancement (10 points)**

Enhance the program that you wrote for Programming Challenge 20 so it keeps a count of the number of guesses that the user makes. When the user correctly guesses the random number, the program should display the number of guesses it took for them to get it right.

#include <iostream>

#include <fstream>

#include <stdlib.h>

#include <time.h>

using namespace std;

int main()

{

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

unsigned seed = time(0);

srand(seed);

cout << endl << "The computer will generate a secret" << endl;

cout << "random number between 1-100. Try to guess" << endl;

cout << "what the number is!" << endl << endl;

z = rand() % 100 + 1;

do

{

cout << "Your guess: ";

cin >> x;

cout << endl;

y++;

if (x < z)

{

cout << "You need to guess higher!" << endl;

}

if (x > z)

{

cout << "You need to guess lower!" << endl;

}

}while (x != z);

cout << endl << "Congratulations! You guessed the secret number: " << z << "!" << endl;

cout << "It took you " << y << " attempts!" << endl;

return 0;

}

1. **Star Search (25 points)**

A particular talent competition has five judges, each of whom awards a score between 0 and 10 to each performer. Fractional scores, such as 8.3, are allowed. A performer’s final score is determined by dropping the highest and lowest score received, then averaging the three remaining scores. Write a program that uses this method to calculate a contestant’s score. It should include the following functions:

* Void getJudgeData () should ask the user for a judge’s score, store it in a reference parameter variable, and validate it. This function should be called by main once for each of the five judges.
* Void calScore () should calculate and display the average of the three scores that remain after dropping the highest and lowest scores the performer received. This function should be called just once by main and should be passed the five scores.

The last two functions, described below, should be called by calScore, which uses the returned information to determine which of the scores to drop.

* int findLowest () should find and return the lowest of the five scores passed to it.
* int findHighest () should find and return the highest of the five scores passes to it.

#include <iostream>

#include <fstream>

#include <stdlib.h>

#include <time.h>

using namespace std;

void judgeData(float&, int&);

void calScore(float, float, float, float, float);

int findLowest(float, float, float, float, float);

int findHighest(float, float, float, float, float);

int main()

{

int x(0);

float jOne(0), jTwo(0), jThree(0), jFour(0), jFive(0);

cout << endl << "Welcome to 'Who's Got Talent?'!" << endl << endl;

cout << "Please enter the scores (1-10) that" << endl;

cout << "the five judges gave!" << endl;

judgeData(jOne, x);

judgeData(jTwo, x);

judgeData(jThree, x);

judgeData(jFour, x);

judgeData(jFive, x);

calScore(jOne, jTwo, jThree, jFour, jFive);

return 0;

}

void judgeData(float& judgement, int& x)

{

x++;

do

{

cout << endl << "Judge " << x << ": ";

cin >> judgement;

if (judgement < 0 || judgement > 10)

{

cout << "You must enter a value 1-10" << endl;

}

}while (judgement < 0 || judgement > 10);

}

void calScore(float jOne, float jTwo, float jThree, float jFour, float jFive)

{

int lowest(0), highest(0);

float total(0), average(0);

lowest = findLowest(jOne, jTwo, jThree, jFour, jFive);

highest = findHighest(jOne, jTwo, jThree, jFour, jFive);

total = jOne + jTwo + jThree + jFour + jFive;

if (lowest == 1 || highest == 1)

{

total -= jOne;

}

if (lowest == 2 || highest == 2)

{

total -= jTwo;

}

if (lowest == 3 || highest == 3)

{

total -= jThree;

}

if (lowest == 4 || highest == 4)

{

total -= jFour;

}

if (lowest == 5 || highest == 5)

{

total -= jFive;

}

average = total / 3;

cout << "The performer's average score is: " << average << endl;

}

int findLowest(float jOne, float jTwo, float jThree, float jFour, float jFive)

{

if (jOne <= jTwo && jOne <= jThree && jOne <= jFour && jOne <= jFive)

{

return 1;

}

else if (jTwo <= jThree && jTwo <= jFour && jTwo <= jFive)

{

return 2;

}

else if (jThree <= jFour && jThree <= jFive)

{

return 3;

}

else if (jFour <= jFive)

{

return 4;

}

else

{

return 5;

}

}

int findHighest(float jOne, float jTwo, float jThree, float jFour, float jFive)

{

if (jFive >= jFour && jFive >= jThree && jFive >= jTwo && jFive >= jOne)

{

return 5;

}

else if (jFour >= jThree && jFour >= jTwo && jFour >= jOne)

{

return 4;

}

else if (jThree >= jTwo && jThree >= jOne)

{

return 3;

}

else if (jTwo >= jOne)

{

return 2;

}

else

{

return 1;

}

}

1. **Days out (25 points)**

Write a program that calculates the average number of days a company’s employees are absent. The program should contain the following functions:

* A function called by main that asks the user for the number of employees in the company. This value should be returned as an int. (the function accepts no arguments.)
* A function called by main that accepts one argument: the number of employees in the company. The function should ask the user to enter the number of days each employee missed during the past year. The total of these days should be retuned as an int.
* A function called by main that takes two arguments: the number of employees in the company and the total number of days absent for all employees during the year. The function should return, as a double, the average number of days absent. (This function does not perform screen output and does not ask the user for input.)

*Input validation: Do not accept a number less than 1 for the number of employees. Do not accept a negative value for the days any employee missed.*

#include <iostream>

#include <fstream>

#include <stdlib.h>

#include <time.h>

using namespace std;

int noEmployees();

int daysAbsent(int);

double calculate(int, int);

int main()

{

int employees(0), absent(0);

double average(0);

cout << endl << endl;

employees = noEmployees();

absent = daysAbsent(employees);

average = calculate(employees, absent);

cout << endl << endl << "The average employee was absent for " << average << " days!" << endl;

return 0;

}

int noEmployees()

{

int employees(0);

do

{

cout << "Please enter the number of employees: ";

cin >> employees;

cout << endl;

if (employees < 1)

{

cout << "Please enter at least one." << endl;

}

}while (employees < 1);

return employees;

}

int daysAbsent(int employees)

{

int absent(0), totalAbsent(0);

for (int x = 1; x <= employees; x++)

{

do

{

cout << "How many days was emplyee #" << x << " abesent? ";

cin >> absent;

if (absent < 0)

{

cout << "Please enter a positive number." << endl;

}

}while (absent < 0);

totalAbsent += absent;

}

return totalAbsent;

}

double calculate(int employees, int absent)

{

double average(0);

average = absent / employees;

return average;

}