Kelly Williams

ECE 1310.04

10/9/2023

Homework 06

Q1: 3.12 *(Gas Mileage)* Drivers are concerned with the mileage obtained by their automobiles. One driver has kept track of sever tankfuls of gasoline by recording miles driven and gallons used for each tankful. Develop a C++ program that uses a **while** statement to input the miles driven and gallons used for each tankful. The program should calculate the miles per gallon obtained for each tankful and print the combined miles per gallon obtained for all tankfuls up to this point.

A paper with text and numbers

Description automatically generated

A screenshot of a computer program

Description automatically generated

/\* Pseudocode

\*

\* decalre doubles for miles, gallons, mpg, sum, and mpgTot

\* int count

\*

\* while miles > 0

\* prompt user to input miles driven and gallons used (-1 to quit)

\* mpg = miles/gallons

\* count++

\* sum = sum + mpg

\* mpgTot = sum/count

\* then output the mpg for this tankful and the total average mpg

\*/

//System Libraries

#include <iostream>

#include <iomanip> //setprecision

//Function Prototypes

//Global Constants

//Main Function

using namespace std;

int main(int argc, char\*\* argv)

{

//declare variables

double miles = 0;

double mileTot = 0;

double galTot = 0;

double mpg, gals, mpgTot;

while (miles >= 0)

{

cout << "Enter miles driven (-1 to quit): ";

cin >> miles;

if (miles == -1)

break;

cout << "Enter gallons used: ";

cin >> gals;

//now for the math

//calculate mpg

mpg = miles / gals;

//get total miles

mileTot = mileTot + miles;

//get total gas used

galTot = galTot + gals;

//divide the sum by the count to get the avg mpg total

mpgTot = mileTot / galTot;

//output results

cout << fixed << "MPG this tankful: " << setprecision(6) << mpg << endl

<< "Total MPG: " << setprecision(6) << mpgTot << endl << endl;

}

return 0;

}

Q2: 3.18 *(Tabular Output)* Write a C++ program that uses a **while** statement and the tab escape sequence **\t** to print the following table of values:

A white paper with black numbers

Description automatically generated

A screenshot of a computer

Description automatically generated

//Main Function

using namespace std;

int main(int argc, char\*\* argv)

{

int N = 1;

cout << "N\t10\*N\t100\*N\t1000\*N\n\n";

while (N <= 5)

{

cout << N << "\t" << 10 \* N << "\t" << 100 \* N << "\t" << 1000 \* N << endl;

N++;

}

return 0;

}

Q3: 3.20 *(Validating User Input)* The examination-results program of Fig. 3.12 assumes that any value input by the user that’s not a 1 must be a 2. Modify the application to validate its inputs. On any input, if the value entered is other than 1 or 2, keep looping until the user enters a correct value.

A screenshot of a computer program

Description automatically generated

//Main Function

using namespace std;

int main(int argc, char\*\* argv)

{

//initializing variables in declarations

int passes = 0; // number of passes

int failures = 0; // number of failures

int studentCounter = 1; // student counter

int result; // one exam result (1 = pass, 2 = fail)

// process 10 students using counter-controlled loop

while (studentCounter <= 10)

{

// prompt user for input and obtain value from user

cout << "Enter result (1 = pass, 2 = fail): ";

cin >> result; // input result

//these are my changes \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

if (result != 1 && result != 2)//this will basically act as a flag to trigger the while loop for validation

{

while (result != 1 && result != 2)//while invalid inputs are given

{

cout << "Invalid input.\n\n"

<< "Enter result (1 = pass, 2 = fail): ";

cin >> result;

} //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

}

//if...else nested in while

if (result == 1) // if result is 1,

passes = passes + 1; // increment passes;

else //else result is not 1, so

failures = failures + 1; // increment failures

//increment studentCounter so loop eventually terminates

studentCounter = studentCounter + 1;

} // end while

//termiation phase; display umber of passes and failures

cout << "Passed " << passes << "\nFailed " << failures << endl;

//determine whether more than eight students passes

if (passes > 8)

cout << "Bonus to instructor!" << endl;

} // end main