**CSC 1101 – Problem Solving and Programming Laboratory**

**Lab 4 – Trevor Trusty**

**25 points – Due January 28, 11pm**

**a)** Save this document with your name and the homework number somewhere in the file name.

**b)** Type/paste your answers into the document.

**c)** Submit this document and your .cpp file(s) to the Canvas item where you downloaded this document. Do not submit a zip file but individually attach your files.

**1) [8 points]** You've been hired by *Olympic Oafs* to write a C++ console application that displays the following world records for speedskating:

|  |  |  |  |
| --- | --- | --- | --- |
| Skater (m-men; w-women) | Country | Event (meters) | Time (minutes:seconds) |
| Pavel Kulizhnikov (m) | Russia | 500 | 0:33.98 = 0.00943888889hours |
| Shani Davis (m) | USA | 1,000 | 1:06.42 = 0.01845 hours |
| Lee Sang-hwa (w) | South Korea | 500 | 0:36.36 = 0.0101 hours |
| Nao Kodaira (w) | Japan | 1,000 | 1:12.09 = 0.020025 hours |



Write **cout** statements to display the data as shown here. Add one more column of calculated data that shows the average speed of each record in kilometers per hour. Use only escape sequences to print the data and to separate it into the five columns. Insure that the data is aligned below the column headers. For a new line, you may use either **endl** or **\n**.



If possible, format your code like this:

Font “Courier New”

Font size “9”

Bold

//==========================================================

//

// Title: World Records

// Course: CSC 1101

// Lab Number: Lab04-01

// Author: Trevor Trusty

// Date: 1/28/2019

// Description:

// Displays world records in Olympic speedskating for 4 skaters

//

//==========================================================

#include <conio.h> // For function getch()

#include <cstdlib> // For several general-purpose functions

#include <fstream> // For file handling

#include <iomanip> // For formatted output

#include <iostream> // For cin, cout, and system

#include <string> // For string data type

using namespace std; // So "std::cout" may be abbreviated to "cout"

int main()

{

// Declare variables

// Show application header

cout << "Welcome to World Records!" << endl;

cout << "--------------------------" << endl << endl;

//Column headings

cout << "Skater (m-men; w-women)\t";

cout << "\tCountry\t";

cout << "\tEvent (meters)\t";

cout << "Time (minutes:seconds)";

cout << "\tAverage Speed" << endl;

cout << "------------------------------------------------";

cout << "-----------------------------------------------------" << endl;

//Row 1

cout << "Pavel Kulizhnikov (m)\t";

cout << "\tRussia\t";

cout << "\t500\t\t";

cout << "0:33.98";

cout << "\t\t\t52.972km/h" << endl;

//Row 2

cout << "Shani Davis (m)\t";

cout << "\t\tUSA\t";

cout << "\t1,000\t\t";

cout << "1:06.42";

cout << "\t\t\t54.200km/h" << endl;

//Row 3

cout << "Lee Sang-hwa (w)\t";

cout << "\tSouth Korea";

cout << "\t500\t\t";

cout << "0:36.36";

cout << "\t\t\t49.504km/h" << endl;

//Row 4

cout << "Nao Kodaira (w)\t";

cout << "\t\tJapan\t";

cout << "\t1,000\t\t";

cout << "1:12.09";

cout << "\t\t\t49.937km/h" << endl;

// Show application close

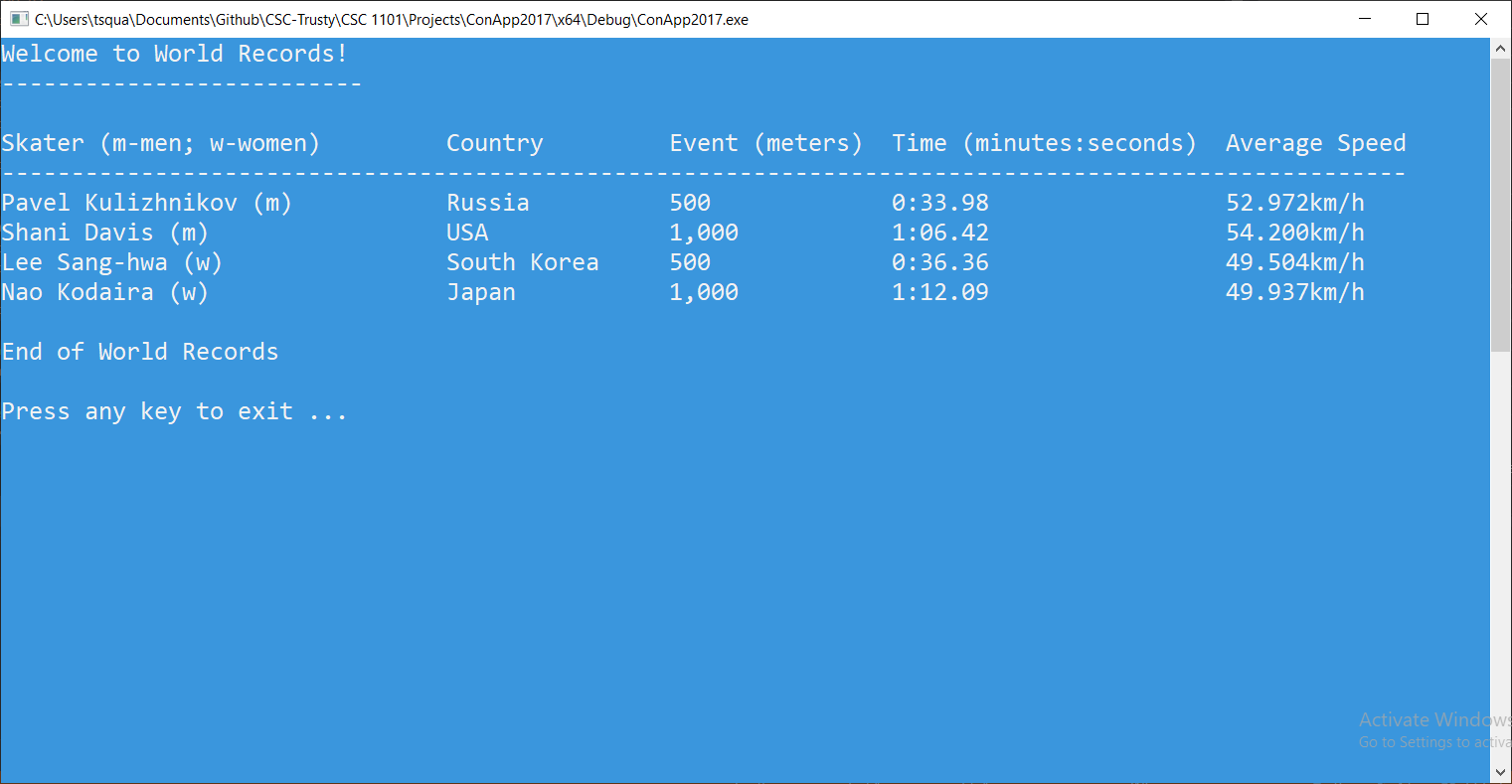
cout << "\nEnd of World Records" << endl << endl;

// Pause before application window closes

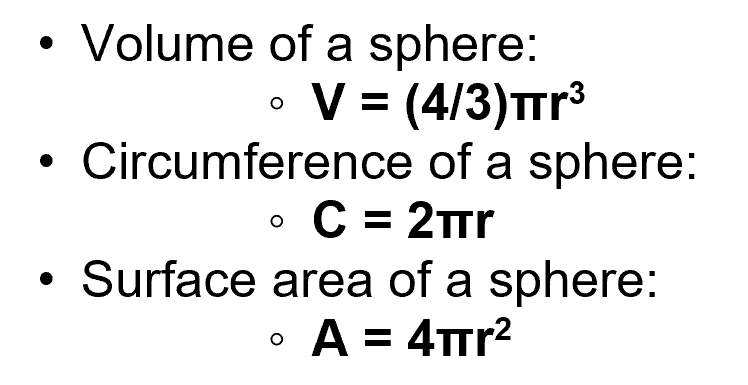
cout << "Press any key to exit ..." << endl;

\_getch();

}



**2) [8 points]** You've been hired by *Spherical Squids* to write a C++ console application that prompts for and gets from the user a real-number radius in centimeters, and calculates the following values:



Declare a constant for π with a value of 3.14159. Use formatted output manipulators (setw, left/right) to print the following four rows:

● Radius

● Volume

● Circumference

● Surface area

And three columns:

● A left-justified label.

● A right-justified value.

● A left-justified units (place one space before the units name to separate it from the second column).

Note that the division operator needs a real number on at least one side to give an accurate result. Format all real numbers to three decimal places. Don't use escape sequences to print the data. The output should look like this:

Welcome to Spherical Squids

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

Enter the radius in cm: 4

Radius: 4.000 cm

Volume: 268.083 cm^3

Circumference: 25.133 cm

Surface area: 201.062 cm^2

End of Spherical Squids

Run the program three times with different values for the radius. What are the results?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Run | Radius | Volume | Circumference | Surface area |
| 1 | 5 | 523.598 cm^3 | 31.416 cm | 314.159 cm^2 |
| 2 | 6 | 904.778 cm^3 | 37.699 cm | 452.389 cm^2 |
| 3 | 7 | 1436.754 cm^3 | 43.982 cm | 615.752 cm^2 |

//==========================================================

//

// Title: Sphere Calculator

// Course: CSC 1101

// Lab Number: Lab04-02

// Author: Trevor Trusty

// Date: 1/28/2019

// Description:

// User enters a radius for a sphere, and console calculates and

// displays the volume, circumfrence, and surface area of the sphere

// with user's radius.

//

//==========================================================

#include <conio.h> // For function getch()

#include <cstdlib> // For several general-purpose functions

#include <fstream> // For file handling

#include <iomanip> // For formatted output

#include <iostream> // For cin, cout, and system

#include <string> // For string data type

using namespace std; // So "std::cout" may be abbreviated to "cout"

int main()

{

//Declare constant

const double pi = 3.14159;

const double f = 4. / 3;

//Variables

int col1 = 16; //Column width

int col2 = 10; //Column width

//int col3 = 0; //Column width

double r;

//Program Header

cout << "Welcome to Sphere Calculator!" << endl;

cout << "--------------------------" << endl << endl;

//Get radius value from user

cout << "Enter the radius in cm: ";

cin >> r;

cout << endl;

//Calculations of Radius

double v = (f)\* pi \* r \* r \* r;

double c = 2 \* pi \* r;

double a = 4 \* pi \* r \* r;

cout << fixed << setprecision (3);

//Output table of values

cout << setw(col1) << left << "Radius:";

cout << setw(col2) << right << r;

cout << left << " cm" << endl;

cout << setw(col1) << left << "Volume:";

cout << setw(col2) << right << v;

cout << left << " cm^3" << endl;

cout << setw(col1) << left << "Circumference:";

cout << setw(col2) << right << c;

cout << left << " cm" << endl;

cout << setw(col1) << left << "Surface Area:";

cout << setw(col2) << right << a;

cout << left << " cm^2" << endl;

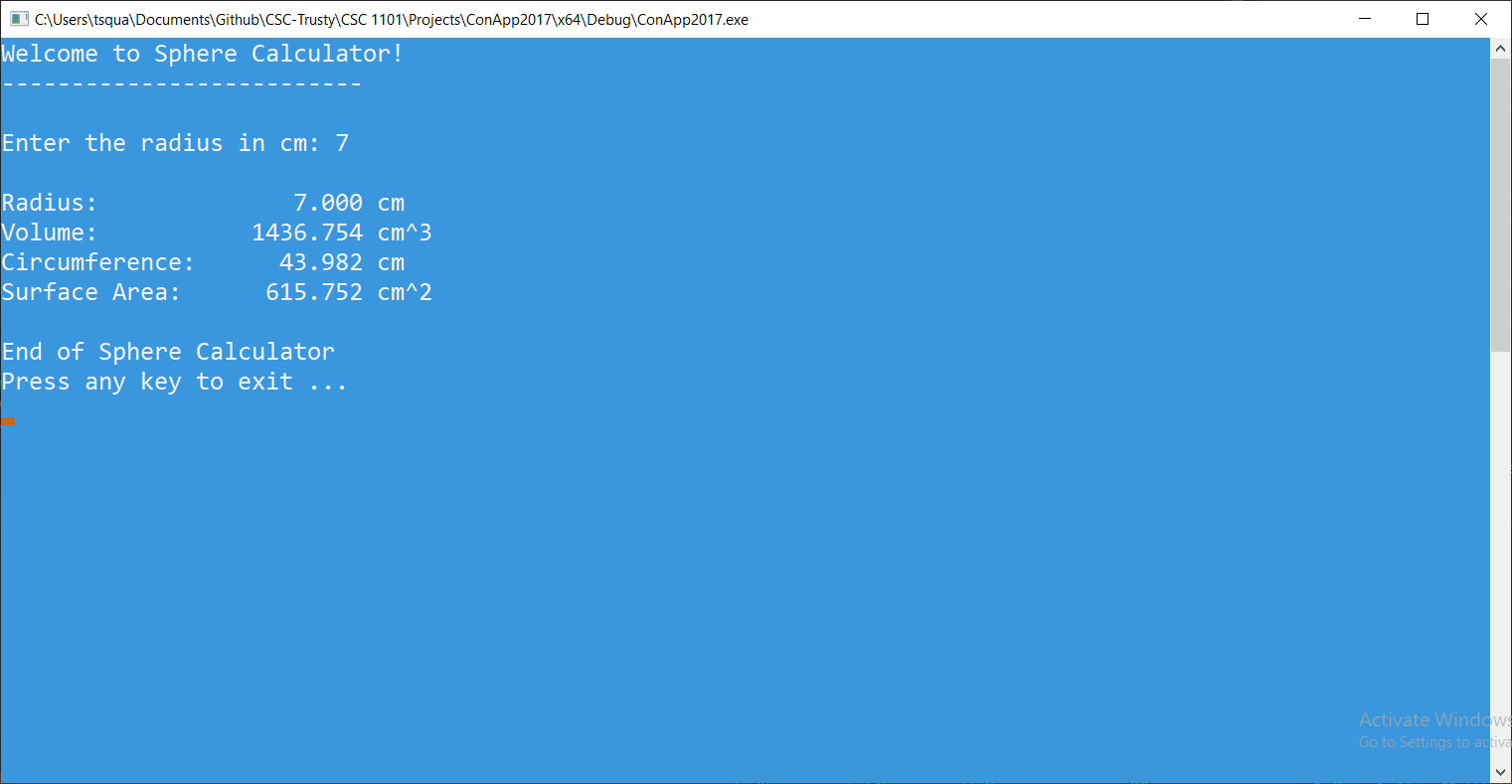
// Pause before application window closes

cout << "\n\aEnd of Sphere Calculator" << endl;

cout << "Press any key to exit ..." << endl;

\_getch();

}



**3) [9 points]** You've been hired by *Gas Gems* to write a C++ console application that shows the estimated cost to drive from one city to another. Prompt for and get from the user the following four inputs:

● City traveling from

● City traveling to

● Distance in miles between the two cities

● Price per gallon of gas

Calculate the gas cost, snack cost, and total cost. For the gas cost, assume the car gets 25 miles per gallon. For the snack cost, assume the car stops every 100 miles and the people spend $7.50 per stop. Declare constants for the miles per gallon (25), snack stop (100), and snack cost ($7.50). The total cost is the sum of the gas cost and the snack cost. Use formatted output manipulators (setw, left/right) to print the following ten rows:

● City traveling from

● City traveling to

● Distance in miles between the two cities

● Price per gallon of gas

● Miles per gallon

● Snack stop miles

● Snack cost per stop

● Gas cost

● Snack cost

● Total cost

And two columns:

● A left-justified label including units if necessary.

● A right-justified value.

Format all real numbers to two decimal places. Don't use escape sequences to print the data. The output should look like this:

Welcome to Gas Gems

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

Enter city traveled from: Detroit

Enter city traveled to: Chicago

Enter distance to travel (miles): 250

Enter price per gallon ($): 2.09

From city: Detroit

To city: Chicago

Distance (miles): 250

Price per gallon ($): 2.09

Miles per gallon: 25.00

Snack stop miles: 100

Snack cost per stop ($): 7.50

Gas cost ($): 20.90

Snack cost ($) 15.00

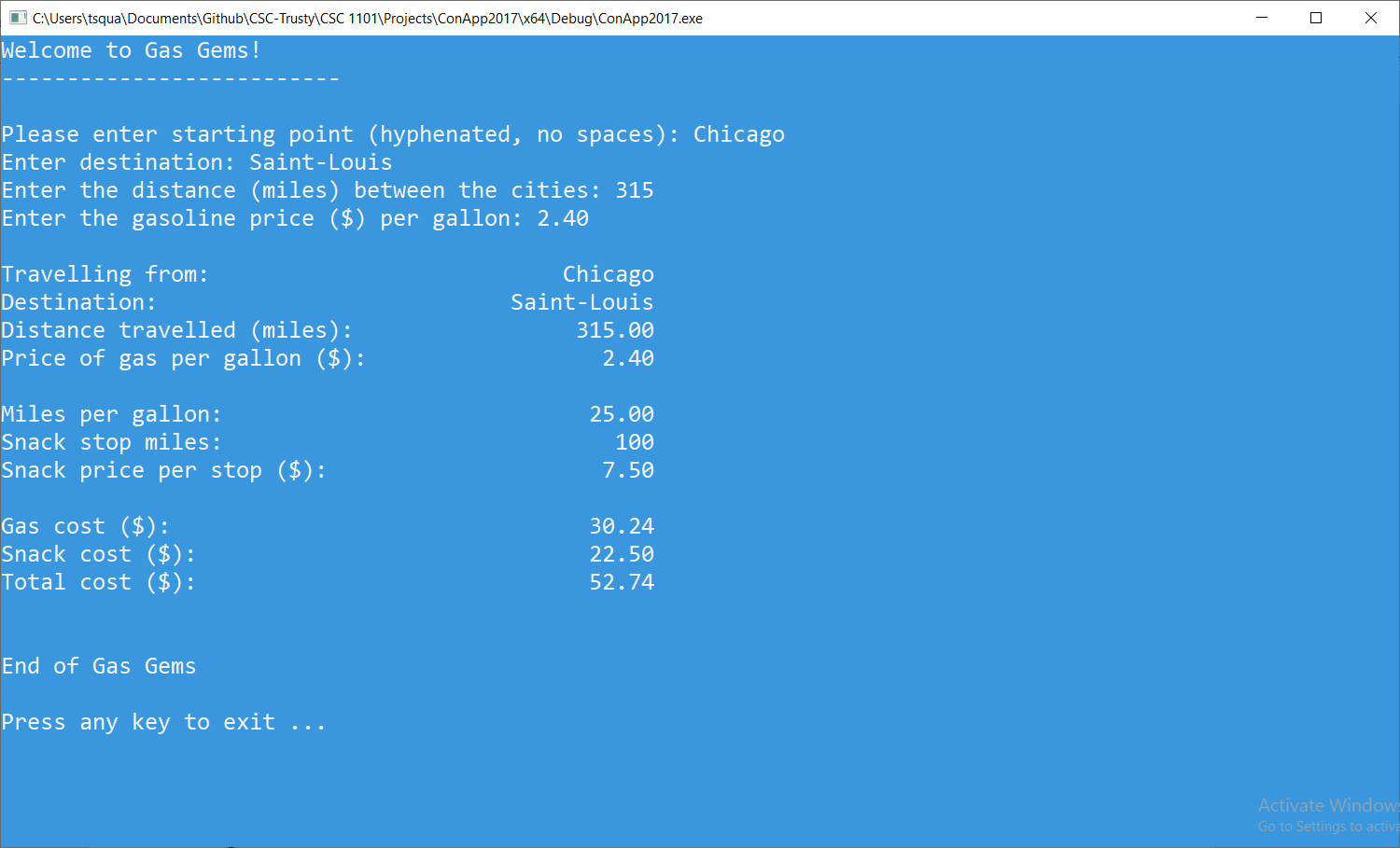
Total cost ($): 35.90

End of Gas Gems

Run the program three times with different values for the inputs. What are the results?

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Run | Distance | Price per gallon | Gas cost | Snack cost | Total cost |
| 1 | *100* | *2.00* | 8.00 | 7.50 | 15.50 |
| 2 | *90* | *1.65* | 5.94 | 0.00 | 5.94 |
| 3 | *315* | *2.40* | 30.24 | 22.50 | 52.74 |

*[your program code here]\**



*[your program output here]\*\**

\* **Copying-and-pasting Visual C++ code to a Word document**

1) From within the Visual C++ program, press **CTRL-A** and press **CTRL-C**.

2) From within the Word document, press **CTRL-V**.

\*\* **Copying-and-pasting Visual C++ console application output to a Word document**

1) From the Visual C++ console, press **ALT-PrintScreen**.

2) From within the Word document, press **CTRL-V**.