**HOMEWORK 6 – A Review of C++**

Answer the following questions. **THIS ASSIGNMENT WILL BE DUE AT A DATE TO BE ANNOUNCED IN CLASS.** (points as noted – 100 points total)

1. Write the code (IN C++!!!) to have the user enter a number, then print out “BINGO” if the number is negative AND a multiple of 8, “dud . . . ” otherwise. Prompt the user if he/she wishes to enter another number. (25 points)

#include <iostream>

using namespace std;

int main() {

float num;

char choice;

do {

cout << "Enter a number: ";

cin >> num;

if ((int)num == (float)num) {

if (int(num) < 0 && int(num) % 8 == 0){

cout << "BINGO!" << endl;

} else {

cout << "dud ..." << endl;

}

} else {

cout << "dud ..." << endl;

}

cout << "Do you want to enter another number? (Y/N): ";

cin >> choice;

} while (choice == 'Y' || choice == 'y');

return 0;

}

1. Write the code to have the user enter 10 doubles into an array, then compute the average of these 10 values. USE SYMBOLIC CONSTANTS!!! Format the average to display 2 decimal places. (25 points)

#include <iostream>

#include <iomanip>

using namespace std;

const int SIZE = 10; // symbolic constant for the size of the array

int main() {

double arr[SIZE];

double sum = 0.0, average;

// prompt the user to enter 10 doubles and store them in the array

for (int i = 0; i < SIZE; i++) {

cout << "Enter value " << i+1 << ": ";

cin >> arr[i];

sum += arr[i];

}

// calculate the average of the values in the array

average = sum / SIZE;

// display the average with 2 decimal places

cout << fixed << setprecision(2) << "The average is: " << average << endl;

return 0;

}

1. Write the code to compute the area of a circle of radius r. You MUST use a function that returns this area; also, you must use a symbolic constant for PI. Finally, you must display the area with two decimal places *with appropriate units*. (25 points)

#include <iostream>

#include <iomanip>

using namespace std;

const double PI = 3.14159; // symbolic constant for PI

double areaOfCircle(double r) {

return PI \* r \* r;

}

int main() {

double radius, area;

cout << "Enter the radius of the circle: ";

cin >> radius;

area = areaOfCircle(radius);

cout << fixed << setprecision(2) << "The area of the circle is: " << area << " sq. units" << endl;

return 0;

}

4. Given the following code:

#include <iostream>

#include <cmath>

using namespace std;

struct Location {

double xc;

double yc;

double zc;

};

struct Point {

int idNum;

double distance0; // distance from ORIGIN

double distance1; // distance from KEYPOINT

struct Location coords;

};

const int NUMPOINTS = 3;

const Location ORIGIN = {0.0, 0.0, 0.0};

const Location KEYPOINT = {11.0, 3.0, 4.9};

void getData(struct Point &P, int i);

float getDistance(struct Location P1, struct Location P2);

int main(void) {

int i = 0;

struct Point myPoints[NUMPOINTS];

for(i=0; i<NUMPOINTS; i++)

getData(myPoints[i],i);

}

void getData(struct Point &P, int i) {

cout << "Now entering data for point " << i+1 << ":\n";

P.idNum = i+1;

cout << "\tEnter x-coordinate: "; cin >> P.coords.xc;

cout << "\tEnter x-coordinate: "; cin >> P.coords.yc;

cout << "\tEnter x-coordinate: "; cin >> P.coords.zc;

P.distance0 = getDistance(P.coords,ORIGIN);

P.distance1 = getDistance(P.coords,KEYPOINT);

}

float getDistance(struct Location P1, struct Location P2)

{

float temp1 = pow((P1.xc-P2.xc),2);

float temp2 = pow((P1.yc-P2.yc),2);

float temp3 = pow((P1.zc-P2.zc),2);

return(sqrt(temp1+temp2+temp3));

}

Now, assume that the following data is entered at the input prompts for the coordinates of our points: <3,4,5>; <5,5,5>; and <6,7,8>.

1. What value is stored in myPoints[2].idNum? (5 points)

The value stored in myPoints[2].idNum is 3, because the idNum field for myPoints[2] is assigned the value of i+1 in the getData function, where i is 2.

1. What value is stored in myPoints[0].coords.yc? (5 points)

The value stored in myPoints[0].coords.yc is 4, because cin >> P.coords.yc; in the getData function reads the user input for the y-coordinate of myPoints[0] and assigns it to the yc field of P.coords.

c. Assume that the following code is added at the end of main:

for (i=0; i<NUMPOINTS; i++) {

cout << myPoints[i].distance0 << "\t";

cout << myPoints[i].distance1 << "\t";

cout << endl;

}

What is the output of this code? (15 points)

7.07107 8.06288

8.66025 6.32535

12.2066 7.11407

This is because the distance0 and distance1 fields of each point in the myPoints array are calculated using the getDistance function with respect to ORIGIN and KEYPOINT, respectively. The output displays these distances for each point.