Object Oriented Programming: Programming Assignment

You must use the template to do this assignment.

**You do not receive any score points if you do not use the template.**

In this assignment, you are going to implement some classes which are integrated in a large system. You can see the files in 00\_StudentWork. There are three classes:

mySystem\_MonteCarlo

mySystem\_QuarticFunction

mySystem\_StudentManager

**Write your programs in .NET2010. You must use the assignment template to implement your programs.**

**We will rebuild your program in the Release mode and check your program.**

**You can find the executable file**

**enjoy\_programming.exe**

**in ./bin/Release**

**The demo program may have bugs. Thus, please follow the instructions. If the instructions are not clear to you, send us an email to clarify the issues.**

Requirement Specification: All the calculations should be done in double precision. Define a variable as double type for floating point computation. Don’t use float. Show the value of a floating point number up to 8 decimal digits.

1. **Basic tasks.**
2. **Write your name in the header file mySystemApp.h**
3. Press ‘s’ or ‘S’ to show your student information: **date**, student ID, name, and email address. showMyStudentInfo\_2020\_Summer( ) in mySystemApp.cpp
4. Set STUDENT\_INFO for your name and student ID in mySystemApp.cpp.

**Items I, II and III must be done. If not, your score is zero.**

Key usages:

F1: perform Monte Carlo Simulation

F3: perform Quatic Function Calculation

F4: perform the student record management

i, I: ask for input

s, S: show the student information

**In mySystem\_MonteCarlo, implement the followings.**

1. Show a message about the Monte Carlo simulation.
2. Press ‘i’ or ‘I’ to ask the user to input the radius of a circle. The radius is inside [1,10]. Ask the user to input the number of samples. The number of samples is between 1 and 1,000,000.
3. Press ‘1’ to set the option to MCS\_OPTION\_CIRCLE. Use mOption.
4. Press ‘2’ to set the option to MCS\_OPTION\_FUNC\_01. Use mOption.
5. Use the Monte Carlo simulation to estimate the area of the region. Show the estimated area value. Use generateUniformSample ( ) to generate samples. Use computeValue( ) to compute the area of the region. Use mValue to store the area.
6. Implement getValue( )
7. Get the radius getRadius( ).
8. Get the number of samples getNumSamples( ).
9. Implement isInsideRegion( )
10. Implement getSample( … ).
11. Implement reset( ) to recompute the samples and the area of the region.
12. Get the coordinates of a sample based on the sample index (starting from 0). Return true if the sample lies inside the region. Otherwise return false.
13. Press ‘<’ to decrease the number of sampler points by 5000 each time. Compute the new samples. Implement method decrease( )**. The minimum number of sample points is 2.**
14. Press ‘>’ to increase the number of sampler points by 5000 each time. Compute the new samples. Implement method increase( ). **The maximum number of sample points is 1,000,000.**
15. Press ‘n’ to decrease the radius of the circle by 1. The smallest radius is 1. Compute the new samples and output the area of the region**.** Implement method decreaseRadius( )**.**
16. Press ‘m’ to increase the radius of the circle by 1. The largest radius is 10. Compute the new samples and output the area of the region**.** Implement method increaseRadius( )**.**

The center of the circle is at the origin, i.e., (0,0).

The region is defined as follows:

When Option = MCS\_OPTION\_CIRCLE, the region is inside the circle.

When Option = MCS\_OPTION\_FUNC\_01, the region is inside the circle and also a point (x,y) is inside the region iff the condition y >= sin(x) + cos(x) is true.

**In mySystem\_QuarticFunction, implement the following items.**

1. Press ‘i’ to ask the user to input the range of x, i.e., minimum value and maximum value of x. The minimum and maximum values should be in the interval [-100, 100]. Ask the user to input the number of sample points inside the interval [2, 500].
2. Get the range of x, i.e., getRangeOfX.
3. Get the number of sample points, i.e., getNumOfSamples.
4. Get the value of the function for a given x value, i.e., getValue.
5. Press ‘1’, set a = 1.0\*k, b = -1.0\*k, c = 2\*k, d = 15\*k, e = 100.0\*k.
6. Press ‘2’, set a = -1.0\*k, b = -5.0\*k, c = 2\*k, d = 15\*k, e = 0\*k;.
7. Press ‘3’, set a = 1.0\*k, b = 1.0\*k, c = -20\*k, d = 15\*k, e = 300.0\*k.
8. Press ‘4’, set a = 1.0\*k, b = 5.0\*k, c = 20\*k, d = -35\*k, e = -100\*k;
9. Press ‘v’ to decrease value b by 0.0002.
10. Press ‘b’ to increase value b by 0.0002.
11. Press ‘n’ to decrease value c by 0.002.
12. Press ‘m’ to increase value c by 0.002.
13. Press ‘,’ to decrease value d by 0.02.
14. Press ‘.’ to increase value d by 0.02.

k = 0.002.

The quartic function is: f(x) = ax4 + bx3 + cx2 + dx + e

**Set the initial values to a, b, c, d, and e, e.g., c = d = 0.5; a = 1.0; b = -0.5; e = 0.0;**

Play with the system by changing the coefficient values, i.e., b, c, and d. See how the curve of the function change accordingly. Do you observe some interesting patterns?

**In mySystem\_StudentManager, implement the following items.**

1. Press ‘i’ or ‘I’ to ask for input.
2. Ask the user to input the number of students. The number of students is in [2,100].
3. Ask the user to input the score of each student. The score range is [0, 100]. The score value is an integer.
4. Show the range of the scores.
5. Show the average score.
6. Show the standard deviation of the scores.

Standard deviation = sqrt( sum(x – x’)\*( x – x’)/(n-1) ), for scores of all students, where x is the score of a student and x’ is the average. Read the article about standard deviation in Wiki if you are not sure what it is.

1. Show the scores in an ascending order. That is, sort the scores.
2. Get the number of students whose score is inside an interval [s0, s1] (inclusive). Implement getNumOfStudentsInScoreInterval(…).

The program shows the histogram of the scores of all the students.

Use ( rand( ))/(double) (RAND\_MAX)

to compute a random value between [0, 1].

**Submission**:

1. Change the folder name to ID\_Name, where ID is your student ID and Name is your name. Upload the entire folder of the source code to E3 platform before the deadline.
2. You must demo your work to our TAs in the lab session.
3. **If you cannot demo your programs, your score is zero.**

**Penalties:**

1. **Late submission: 30% penalty each day.**
2. **Cheating: you will be received a score of zero, e.g., borrowing your source code to others or/and copying others’ source code.**

**The folder 00\_StudentWork stores the files.**

