Object Oriented Programming

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

GALAXY\_SYSTEM

MONTE\_CARLO\_SYSTEM

SIMPLE\_FILE\_EDITOR

SIMPLE\_PAINTER

SORTING\_VISUALIZATION

Write your program on .NET2010. You must use vc100 toolset. Open the project: SOGLFramework.sln. We will rebuild your program in the Release mode and check your program.

The demo program may have bugs. The demo program shows only some basic results. It is incomplete. You must follow the instruction to implement the tasks.

**Submission**:

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

Penalties:

1. Late submission: 40% penalty each day.
2. Cheating: you will be received a score of zero. For examples, borrowing your source code to others or/and copying others’ source code.
3. **You must not add new files. Deduct 50pt if you add new files.**

The folder 00\_StudentWork stores the files.

The binary file is SelfMotivatedSystem.exe which is stored in ./bin/release/.

To run the program: 1) go the folder ./bin/release/. 2) Then execute SelfMotivatedSystem.exe.

**Do not press F5 to run the program. We do not set the file directories in the project. Go to the folder and click** SelfMotivatedSystem.exe to run it.

**Requirement Specification**

All the calculations should be done in double precision in **Monte Carlo Simulation**.

**You can add new functions but you must not add new files. Deduct 50pt if you add new files.**

**Basic tasks.**

1. Write your name, student ID and email address in the header file mySystemApp.h
2. Press ‘s’ or ‘S’ to show your student information: date, student ID, name and email address. Also, **show the course information**. See **showStudentInfo\_2019**( ) in mySystemApp.cpp
3. Set STUDENT\_INFO for your name and student ID in mySystemApp.cpp.

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

Key usages:

F1: perform Monte Carlo Simulation

F2: perform Galaxy Simulation

F3: perform Painter

F4: perform File to read a bitmap from a file

F5: perform visualization of bubble sort

i, I: ask for input (to see other key usages in different systems)

r,R: reset

s, S: show the student information on the console window

In GALAXY\_SYSTEM, implement the followings.

1. void askForInput( )
2. int getNumOfObjs( ) const
3. bool handleKeyPressedEvent( int key )
4. bool getObjInfo( int objInbdex, double &x, double &y, double &r ) const
5. void update( )
6. void reset( )
7. void mergeObjects( )
8. void generateObjects( )
9. and so on.

You can see the details in the header and source files (mySystem\_GalaxySystem).

In MONTE\_CARLO\_SYSTEM, implement the followings.

1. void computeSamples( )
2. void askForInput( )
3. int getNumSamples( ) const
4. void getSample(int sampleIndex, double &x, double &y ) const
5. void getRangeOfX( double &minX, double &maxX ) const
6. double getValue(double x) const
7. bool isInsideArea( double x, double y ) const
8. double computeArea( ) const
9. bool handleKeyPressedEvent( int key )
10. void reset( )
11. and so on.

You can see the details in the header and source files (mySystem\_MonteCarlo).

In SIMPLE\_FILE\_EDITOR, implement the followings.

1. void readFile( )
2. void storeOneRowToBitmap( const std::string &inputString )
3. void askForInput( )
4. void getBitmapDimension(int &nx, int &ny) const
5. int getBitmapValue(int x, int y) const
6. Functions for rotating the image
7. and so on.

You can see the details in the header and source files (mySystem\_SimpleFileEditor).

In SIMPLE\_PAINTER, implement the followings.

1. void drawAt(int x, int y, const vector3 &color, double w )
2. int computeCanvasIndex( int x, int y, int nx, int ny )
3. void reset( )
4. bool handleKeyPressedEvent( int key )
5. void clickAt(double x, double y)
6. void askForInput( )
7. int getBrushSize( ) const
8. void getCanvasDimension( int &nx, int &ny ) const
9. void getColorAtPixel( int x, int y, vector3 &color ) const
10. void setColor( const vector3 &color )
11. void setTransparency( float v )
12. and so on.

You can see the details in the header and source files (mySystem\_SimplePainter).

In SORTING\_VISUALIZATION, implement the followings.

1. void checkAndSwapTwoElements( int index )
2. void reset( )
3. void askForInput( )
4. int getNumOfElements( ) const
5. int getElement( int elementIndex ) const
6. int getMaxNonHandledElementIndex( ) const
7. void processOneSwapStep( )
8. int getCurrentElementIndex( ) const
9. and so on.

You can see the details in the header and source files (mySystem\_SortingVisualization).

Use ( rand( ))/(double) (RAND\_MAX) to compute a random value between [0, 1].

**Project properties**

Please look for some information about project properties, including C/C++ general, Linker general, and Linker include. Learn about additional dependencies, additional library directories, and so on. **Do you know when you can use F5 and F7?**









