CPE 372 Object Oriented Analysis and Design
Exercise 2
Hiding Information

1. Download the following Java source files from the **Lecture2** subdirectory under **demos:**

   ```
   Square.java
   SquareTesterGraphics.java
   DrawingCanvas.java
   FigureViewer.java
   ```

2. Modify `FigureViewer.java` to remove all references to `Triangle.java` and to `BadSquare.java`. Also uncomment the code that moves and redraws squares. Compile and test the resulting code to make sure it works correctly, before you continue – both drawing and moving the squares.

3. In the lecture, I pointed out that even though the data items in `Square.java` have ***private*** visibility, there is still an undesirable amount of coupling between that class and `FigureViewer.java`, which knows how to draw a square. The current design also has some problems because every time we create a new type of shape, we have to add a new method to `FigureViewer.java` to draw that shape.

4. Your first task in this exercise is to modify some or all of the classes above to get rid of the dependency between `Square` and `FigureViewer`. Specifically, I want you to implement a *draw()* method in `Square.java`, so that a square knows how to "draw itself". Thus the `FigureViewer` will not need any information about how data are stored in the `Square` class. Instead, you can simply call *draw()* for the square to be drawn, from inside `SquareTesterGraphics.java` (see my hint below).

5. Your second task is to implement a *drawAllSquares()* method in `Square.java`. This will be a static method; you will call it as follows: `Square.drawAllSquares()`. (You might need some arguments.) Add code to the end of `SquareTesterGraphics.java` to call this method. Remove the `System.exit()` call so the program will not exit until the user clicks the "Exit" button.

   To do this task, you will need to create a static collection where you keep all the squares that are created (which happens any time there is a call to the *Square()* constructor).

   I recommend that you use an `ArrayList` for this. (See Lecture 13 in the CPE111 notes for more information about ArrayLists.)

   http://windu.cpe.kmutt.ac.th/cpe111/Lectures/Lecture13/index.html

6. Do not forget to follow the Java coding standards. In particular, make sure you add your name, nickname, student ID and the date to the header comment at the top of every class you change. If you forget this, you will lose credit.

7. Upload whatever Java classes you have to change in order to do the two tasks above.

Hint: To do drawing, you need a `Graphics2D` instance. This instance is available from `FigureViewer`. You can implement a "getter" function in `FigureViewer` that returns the `Graphics2D` object. Then you will be able to call drawing functions directly from `SquareTesterGraphics.java`.