CPE 372 Object Oriented Analysis and Design
Exercise 3
Class Hierarchies and Abstract Classes

1. Download the following Java source files from the **Lecture3** subdirectory under **demos:**

```
Square.java
Triangle.java
SquareTesterGraphics.java
DrawingCanvas.java
FigureViewer.java
AbstractShape.java
```

2. Modify `Square.java` and `Triangle.java` so that they are subclasses of `AbstractShape.java`. To do this, you should delete the member data items that have been moved to `AbstractShape`. You should also delete the *drawAll()* method, which is implemented in `AbstractShape`. You will also need to change the *draw()* and *move()* methods to work with `Point` objects, and to get their information from the *vertices* list in `AbstractShape.java`.

3. Implement a new class called `Diamond.java`, which should also extend `AbstractShape`. The constructor for `Diamond` should require X and Y coordinates, for the top point of the diamond, plus a vertical axis length and horizontal axis length. Implement all the necessary methods (*draw()*, *move()*, *calcPerimeter()*, *calcArea()*) for a diamond shape.

4. Using `SquareTesterGraphics.java` as a model, create a new class called `ShapeTesterGraphics.java`. This class will allow the user to repeatedly choose a type of shape —Triangle, Square or Diamond—and draw it in the FigureViewer, then print the perimeter and area, then optionally move and redraw it. The class will also have an option to display all shapes of every type. The interaction should look something like the following:

```
 1 – Create and draw triangle
 2 – Create and draw square
 3 – Create and draw diamond
 4 – Draw all shapes
 5 – Exit
Enter your choice:  2

Enter x for upper left corner of square:  10
Enter y for upper left corner of square:  5
Length of each side of square: 6
 Perimeter: 24   Area: 36
Move it? Y
Enter x coordinate of new position: 5
Enter y coordinate of new position: 8
```

After moving a shape, the program should draw it again.

5. The loop should continue until the user chooses option 5. Then it should exit.

6. Be sure to follow the coding standards. In particular, be sure that every member and method has a Javadoc style comment. Make sure your name, ID and the date are in the header comments of all the classes you change or create. Upload `ShapeTesterGraphics.java`, `AbstractShape.java`. `Triangle.java,` `Square.java and Diamond.java`. You may not need to modify `AbstractShape.java,` but upload it anyway.