Abstract Classes: Exercise 1 p.1

1) Write an abstract class Shape
   - Data members: numSides
   - Constructor: initialize numSides
   - Concrete method: get method for numSides
   - Abstract methods: getArea(), getPerimeter()
2) Write a concrete subclass Rectangle
   - Data members: width, height
3) Write a concrete subclass RtTriangle
   - Data members: width, height
4) In another class, write a main method to define a Rectangle and a Triangle.

Abstract Classes: Exercise 1 p.2

1) Write an abstract class Shape
   - Data members: numSides
   - Constructor: initialize numSides
   - Concrete method: get method for numSides
   - Abstract methods: getArea(), getPerimeter()
2) Write a concrete subclass Rectangle
   - Data members: width, height
3) Write a concrete subclass RtTriangle
   - Data members: width, height
4) In another class, write a main method to define a Rectangle and a Triangle.

Abstract Classes: Exercise p.3

1) Write an abstract class Shape
   - Data members: numSides
   - Constructor: initialize numSides
   - Concrete method: get method for numSides
   - Abstract methods: getArea(), getPerimeter()
2) Write a concrete subclass Rectangle
   - Data members: width, height
3) Write a concrete subclass RtTriangle
   - Data members: width, height
4) In another class, write a main method to define a Rectangle and a Triangle.

Abstract Classes: Exercise p.4

1) Write an abstract class Shape
   - Data members: numSides
   - Constructor: initialize numSides
   - Concrete method: get method for numSides
   - Abstract methods: getArea(), getPerimeter()
2) Write a concrete subclass Rectangle
   - Data members: width, height
3) Write a concrete subclass RtTriangle
   - Data members: width, height
4) In another class, write a main method to define a Rectangle and a Triangle.

Interface: Exercise 2 p.1

1) Write an interface Resizable
   – Has a method `resize(double x)` that resizes a Shape's dimensions by factor x
2) Make Rectangle implement Resizable
3) Write a main method to:
   - Define a Rectangle (width = 2, height = 3)
   - Print the Rectangle"s area & perimeter
   - Resize the Rectangle by factor of 2
   - Re-print the Rectangle"s area & perimeter

Interface: Exercise 2 p.2

1) Write an interface Resizable
   - Has a method `resize(double x)` that resizes a Shape’s dimensions by factor x
2) Make Rectangle implement Resizable
3) Write a main method to:
   - Define a Rectangle (width = 2, height = 3)
   - Print the Rectangle�What"s area & perimeter
   - Resize the Rectangle by factor of 2
   - Re-print the Rectangle"s area & perimeter

Interface: Exercise 2 p.3

1) Write an interface Resizable
   – Has a method `resize(double x)` that resizes a Shape’s dimensions by factor x
2) Make Rectangle implement Resizable
3) Write a main method to:
   - Define a Rectangle (width = 2, height = 3)
   - Print the Rectangle‟s area & perimeter
   - Resize the Rectangle by factor of 2
   - Re-print the Rectangle‟s area & perimeter

Polymorphism: Exercise

- Write a main method
  - Create a `Rectangle` and a `RtTriangle`
  - Add them to an `ArrayList` of *`Shape`s*
  - Iterate through the Shapes in the ArrayList
    - If the Shape is Resizable, resize it by a factor of 0.5
    - Print out perimeter and area

1.00 / 1.001 / 1.002 Introduction to Computers and Engineering Problem Solving
Spring 2012