

## Interface

1-) Listed next is the skeleton for a class named City. Each city has a name and temperature:

```
class City {  
    private String cityName;  
    private double temperature;  
}
```

Flesh out the class with appropriate accessors, constructors, and mutators. The temperatures are assigned by you and can be set from outside the City class—your code does not have to ensure that they are unique.

Next, modify the class so that it implements the Comparable interface.

The class also overrides the compareTo method. This method imposes an order between instances of the City class depending upon their temperature.

Test your class by creating an array of sample temperatures and sort them in an ascending order using a sorting method that takes as input an array of type Comparable.

2-) Define an interface named Shape with a single method named area that calculates the area of the geometric shape:

```
public double area();
```

Next, define a class named Circle that implements Shape.

The Circle class should have an instance variable for the radius, a constructor that sets the radius, accessor/ mutator methods for the radius, and an implementation of the area method.

Also define a class named Rectangle that implements Shape. The Rectangle class should have instance variables for the height and width, a constructor that sets the height and width, accessor and mutator methods for the height and width, and an implementation of the area method.

The following test code should then output the area of the Circle and Rectangle objects:

```
public static void main(String[] args)  
{  
    Circle c = new Circle(4); // Radius of 4  
    Rectangle r = new Rectangle(4,3); // Height = 4, Width = 3  
    ShowArea(c);  
    ShowArea(r);  
}  
  
public static void ShowArea(Shape s)  
{  
    double area = s.area();  
    System.out.println("The area of the shape is " + area);  
}
```

3-) Create a Student class that has instance variables for the student's last name and ID number, along with appropriate constructors, accessors, and mutators.

- Make the Student class implement the Comparable interface.
- Define the compareTo method to order Student objects based on the student ID number.
- In the main method, create an array of at least five Student objects, sort them using Collections.sort, and output the students.
- They should be listed by ascending student number.
- Next, modify the compareTo method so it orders Student objects based on the lexicographic ordering of their last name.
- Without modification to the main method, the program should now output the students ordered by name.