1. Define an interface name `Shape` with a single method named `area` that calculates the area of the geometric shape:

   ```java
   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 name `Rectangle` that implements `Shape`. The `Rectangle` class should have instance variables for height and width, a constructor sets the height and width, accessor/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:

```java
public class DriverClass{
    public static void main(String[] args){
        Circle c = new Circle(4);
        Rectangle r = new Rectangle(4,3);
        showArea(c);
        showArea(r);
    } //end main

    public static void showArea(Shape s){
        Double area = s.area();
        System.out.println("The area of the shape is "+area);
    } //end showArea
} //end DriverClass
```
2. In the sport of diving, seven judges award a score between 0.0 and 10.0. The highest and lowest scores are discarded. The remaining scores are added together. The sum is then multiplied by the degree of difficulty between [1.2, 3.8] for that dive. The result is then multiplied by 0.6 to determine the diver’s score. Write a program that inputs the degree of difficulty and the 7 judges’ scores, and outputs the diver’s score. You are required to use an ArrayList of Double to store the scores.

3. Write a program that uses an ArrayList of parameter type Contact to store a database of contacts. The Contact class should store the contact’s first and last name, phone number, and email address. Add appropriate accessor and mutator methods. Your program should present a menu that allows the user to add a contact, display all contacts, search for a specific contact and displays it, or search for a specific contact and give the user the option to delete it. The searches should find any contact where any instance variable contains a target search string. For example, if “elmore” is the search target then any contact where the first name, last name, phone number, or email address contains “elmore” should be returned for display or deletion. Use the “for-each” loop to iterate through the ArrayList.