Surprise Quiz 2

Create a simple interface that allows an object to be saved to some sort of storage medium. The exact type of medium is not known to the interface (nor to the classes that implement it). The interface will just specify 2 methods, one to return an ArrayList of values to be saved and the other to populate the object's fields from an ArrayList.

Create some sample classes that implement your saveable interface (we've used the idea of a game with Players and Monsters, but you can create any type of classes that you want). Override the **toString()** method for each of your classes so that they can be easily printed to enable the program to be tested easier.

In Main, write a method that takes an object that implements the interface as a parameter and "saves" the values. We haven't covered I/O yet, so your method should just print the values to the screen. Also in Main, write a method that restores the values to a saveable object. Again, we are not going to use Java file I/O; instead, use the *readValues*() method below to simulate getting values from a file – this allows you to type as many values as your class requires, and returns an ArrayList.

// interface

```
import java.util.List;

public interface ISaveable {
    ArrayList<String> write();
    void read(List<String> savedValues);

}
```

//*readValues*

```
public static ArrayList<String> readValues() {
        ArrayList<String> values = new ArrayList<String>();

        Scanner scanner = new Scanner(System.in);
        boolean quit = false;
        int index = 0;
        System.out.println("Choose\n" +
```

```
            "1 to enter a string\n" +
            "0 to quit");

    while (!quit) {
        System.out.print("Choose an option: ");
        int choice = scanner.nextInt();
        scanner.nextLine();
        switch (choice) {
            case 0:
                quit = true;
                break;
            case 1:
                System.out.print("Enter a string: ");
                String stringInput = scanner.nextLine();
                values.add(index, stringInput);
                index++;
                break;
        }
    }
    return values;
}
```