**Lab 3 (Inheritance and Polymorphism)**

**Implement the following classes:**

|  |
| --- |
| **User** |
| - username : String - password : String - items[] : String  |
| + User(username : String, password : String, items[] : String)+ setUsername(username : String) : void + getUsername() : String + setPassword(password : String) : void + getPassword() : String + setItems(items[] : String) : void + getltems() : String[] # abstract getAccessibleitems() : String[] + toString() : String  |

|  |
| --- |
| **Admin** |
|  |
| + Admin(username : String, password : String, items[] : String) + getAccessibleitems() : String[] + toString() : String  |

|  |
| --- |
| **AccountingUser** |
|  |
| + AccountingUser(username : String, password : String, items[] : String) + getAccessibleltems() : String[] + toString() : String |

In this assignment, you will be using inheritance and polymorphism to create an admin and an accounting user for some system.

The User class is an abstract class made up of a username, a password, and an array of accessible items. It has setter and getter methods and an abstract method called getAccessibleltems which should be implemented by subclasses of User.

The Admin class and the AccountingUser class are both subclasses of User. This means that you have to use inheritance to create both classes. Each of these classes constructors must call the superclass constructor. Moreover, both of these classes have to implement the getAccessibleltems method which has to return an array of strings describing the items the user has access to.

Usually, a system admin has access to every item, while another user has access for a limited set of items. For example, the AccountingUser will have access to an accounting related item but not any other item.

Write a tester class which has an array of Users.

The array should contain one Admin and two AccountingUsers. After creating the users, you should loop over the array and use polymorphism to get the accessible items and print the user's information.

 Printing format is as follows:

 (username)

Accessible items: (getAccessibleltems())

Solution: <http://www.javaproblems.com/2013/12/creating-inhertiance-application-in-java.html>