SfwrEng/CompSci 2S03 Fall 2015 Homework 4

In a **group of three,** you are to develop a Java program implementing OO Concepts to build the e-Commerce application described below. The system only sells 4 types of items: Books, Ebooks, CDs, and MP3. All details of *Registered Users* and *Products* are stored in a file (see sample output).

Database

All the information is stored in comma-separated files:

- **Usernames:** all the usernames of *Registered Users* are stored in the file **Users.txt.**
- **Items:** All books, ebooks, CDs, and MP3 information are stored in **Books.txt, Ebooks.txt**, **CDs.txt,** and **MP3.txt**, respectively (all case sensitive). Books and ebooks details are: S.No, name of the book, author, price, quantity in Store. CD and MP3 details are: S.No, name of the audio, artist, price, quantity in Store and type (CD or MP3).
- **Shopping cart:** The content of the shopping cart is stored in **Cart[username].txt** with the following fields (S.No, product name, date added, quantity).

Users.txt

```
White
Jack
Brown
```

Cart_[username].txt

```
1, Java Rocks, 20/11/2015, 1
2, Harry Potter All Vol, 19/11/2015, 1
```

Books.txt

```
3, Java Rocks, XYZ, 150, 10
4, Python Sucks, YYZ, 200, 20
```

Ebooks.txt

```
5, Harry Potter All Vol., ZZZ, 200, 5
6, Twilight, DSS, 250, 3
```

CDs.txt

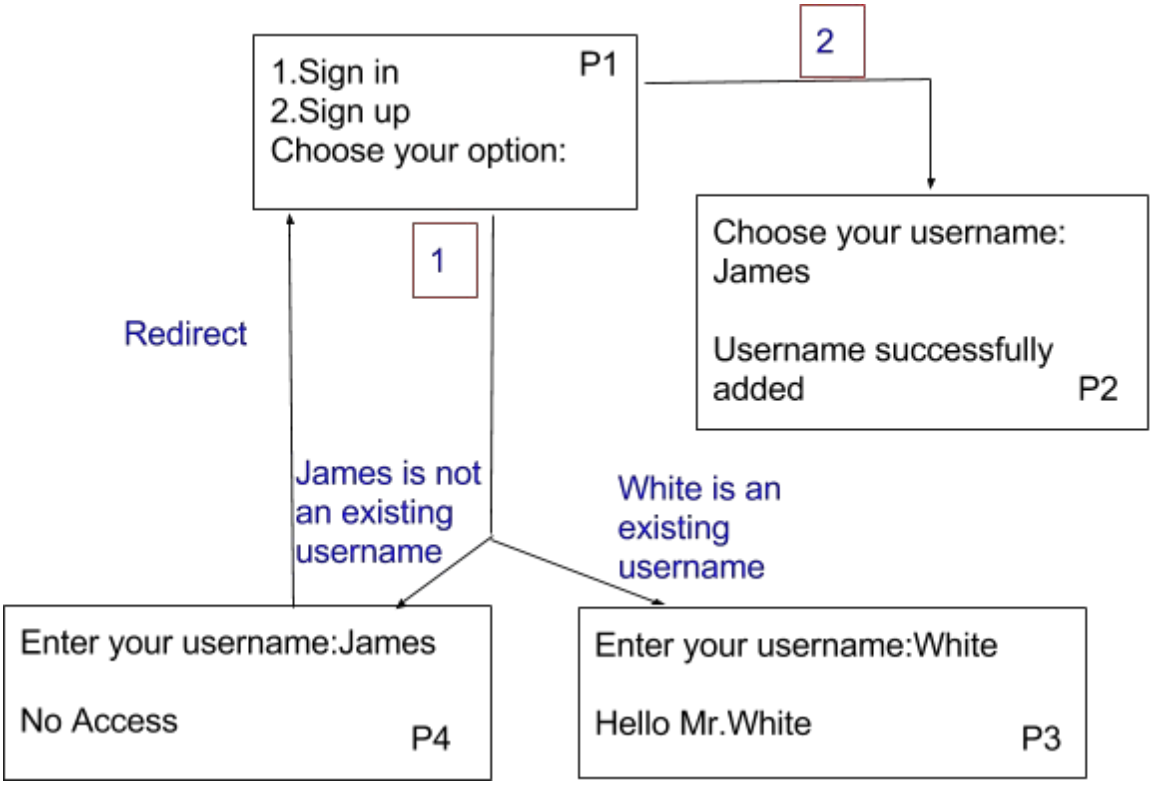
```
7, Java you are in my heart, ADF, 100, 5
8, C++ you are my soul, DEF, 150, 5
```

MP3.txt

```
9, GHID, CDF, 20, 5
10, FTHD, REF, 10, 5
```

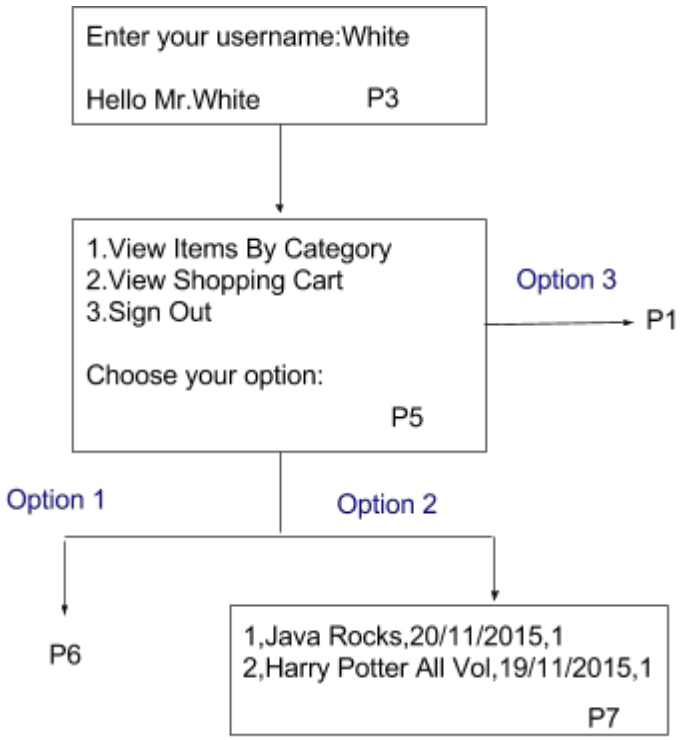
User Interface

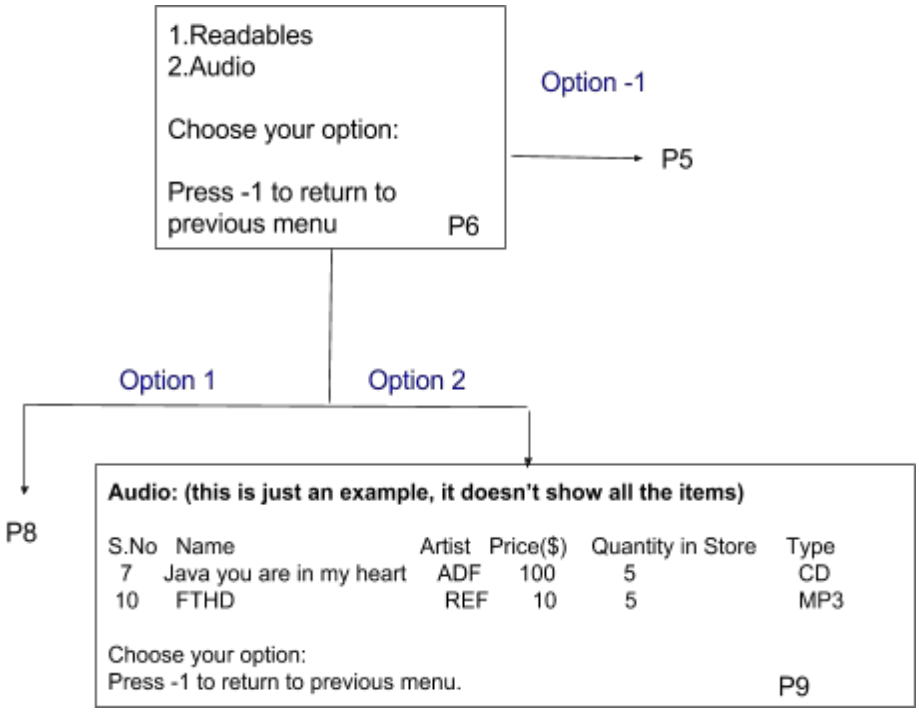
- The execution starts with the login page (P1). If a *Username* already exists in Users.txt, upon signing in, access is granted and the system prints this message “Hello Mr.[Username]” and goes to P5. Otherwise, it prints “No Access” and goes back to P1.

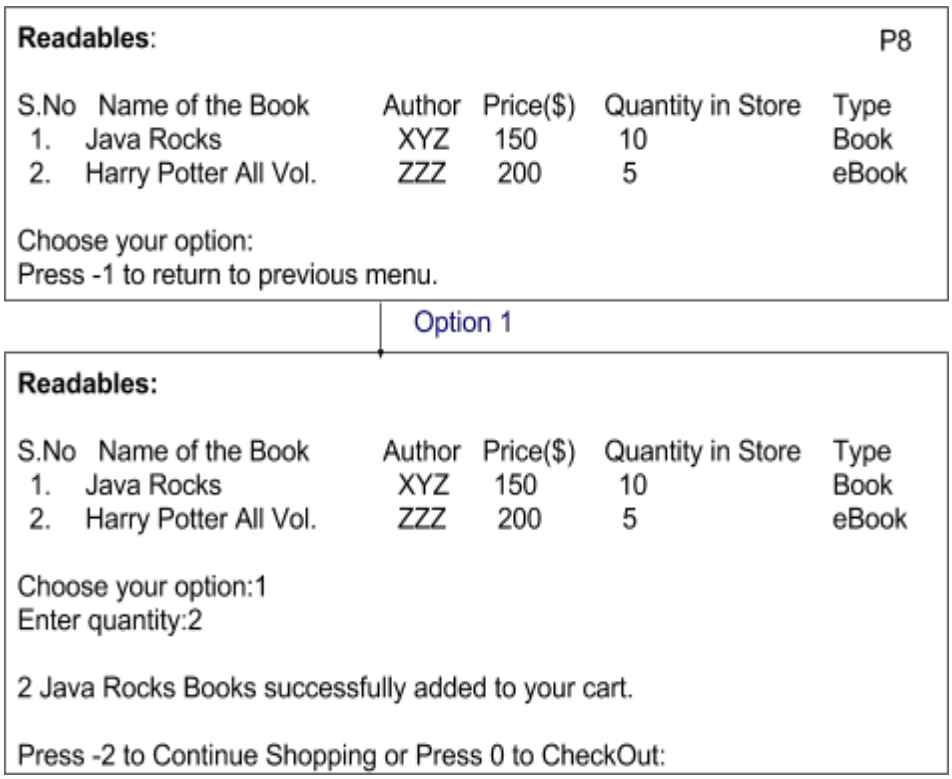


- After Sign In, the following options are displayed (P5). If option 1 is chosen, item categories are listed (P6). If option 2 is chosen, the details of the shopping cart from **Cart[username].txt** is listed (P7) and for Option 3, redirect to *Sign In* page (P1).
- From P6, the following list of options is displayed under View Items By Category. If option 1 is chosen, list all Readables with details (P8). For option 2, list all Audio products (similar to P8). For option -1, the control goes to the previous page (P5).
- From P8, if option 1 or 2 is chosen, you will get the user input on the id of the item and the quantity and print a message saying the selected items have been added to the cart. If the requested quantity is not available, an appropriate message should be printed. If the User prefers to continue shopping, redirect to P6 or else go to Checkout.
- Each time an item is purchased, quantity is deducted and the new quantity is stored in the corresponding .txt file.
- In the Check out (P10), add the list of items in the Cart, apply HST(13%) for every item, **Environment tax (2% on each time), and Shipping (10%) for CDs and Books only** on each item purchased.

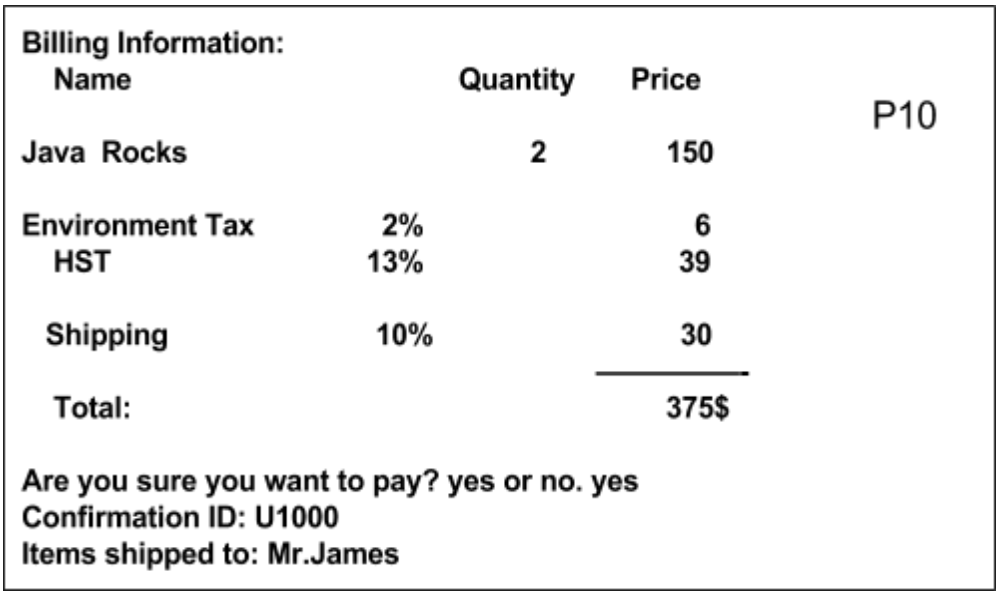
- After prompting the user to check if he wants to pay, print the Confirmation Id (Starting with U1000) and print the name of the person to which the item is shipped and go to the Browse item page.







- If option 0 is chosen go to P10 else if -2 is chosen redirect to P6



- Ignore case sensitivity; i.e. Yes, yes, YES, YeS, yeS, No, no, nO, etc… should all be valid forms of accepting payment.

OO Design

- Your implementation should follow the following structure of classes. You may add new methods to have a better level of modularity.

```
public abstract class Item{
    public abstract String getInfo(...);
    public abstract int getPrice(...);

    protected int price;
    protected int sNo;
// Add other fields if necessary

}
public class Readable extends Item {

    protected String authorName;
    public String getInfo(...){...} //Returns sNo, Name, Author
name, etc in a string
   @Override
   public int getPrice(...){ // override
   …
   }
}
public class Book extends Readable {
…
   @Override
```

```
   public int getPrice(...){ // override to get the item price and
add 2% (Environment Tax)
…
   }
}
   public class eBook extends Readable{
   @Override
   public int getPrice(...){ // override and only call the
parent's constructor to get the base price.
…
}

public class Audio extends Item
{
 protected String artistName;
    public String getInfo(...){...} //Returns sNo, Name, Artist
name, etc in a string
   @Override
   public int getPrice(...){ // override
   …
…
}
public class CD extends Audio {
…
   @Override
   public int getPrice(...){ // override to get the item price
and add 2% (Environment Tax)
…
   }
}
public class MP3 extends Audio{
   @Override
   public int getPrice(...){ // override and only call the
parent's getPrice() to get the base price.
…
}

public class User{
   private String username;
   public String getUsername(...){ // stores the username.
…
}
public class ShoppingCart extends User
{
   private content //array of items
```

```
   public String getContent(...){// return the content of the
shopping cart
   public addItem (...){}
}
public class UserInterface{
   private array readables;
   private array audioProducts;
   private int currentPage; // the page number (P1..P10)
   public int getCurrentPage(...){// This method is for page
navigation. Based on the values of the state variable, call
different pages
   public int changeCurrentPage(...){// This method is for page
navigation. It should change to current page and show the content.
  public void getReadables(); // Fetches all readables from the
files and places them in the readables array
  public void getAudioProducts(); // Fetches all audio products
from the files and places them in the readables array
    public void showReadables(); // Displays all readables for
browsing
    public void showAudioProducts(); // Displays all audio
products for browsing
…
}
```

Submission

*****Only one submission per group*****

All files are to be submitted using the Avenue system. Please ensure you submit all files with the correct names, as described below:

- Upload a **ZIP** file named HWK4_MACID.zip
    - NOTE: the folder zipped must also be named HWK4_MACID

  The ZIP file must contain the following .java source files:

    - HWK4_MACID.java (contains the main method)
    - Item.java
    - Readable.java
    - Book.java
    - eBook.java
    - Audio.java
    - CD.java
    - MP3.java
    - User.java
    - ShoppingCart.java
    - UserInterface.java

- If you don't follow the above naming convention, you will receive no credit.
- If your program does not compile, you will receive 0. Your file must be submitted **by 8:30am** on **Tuesday, November 24th** on Avenue to Learn.

Documentation

Your program must be commented properly: each section of the code as well as each line.

Format

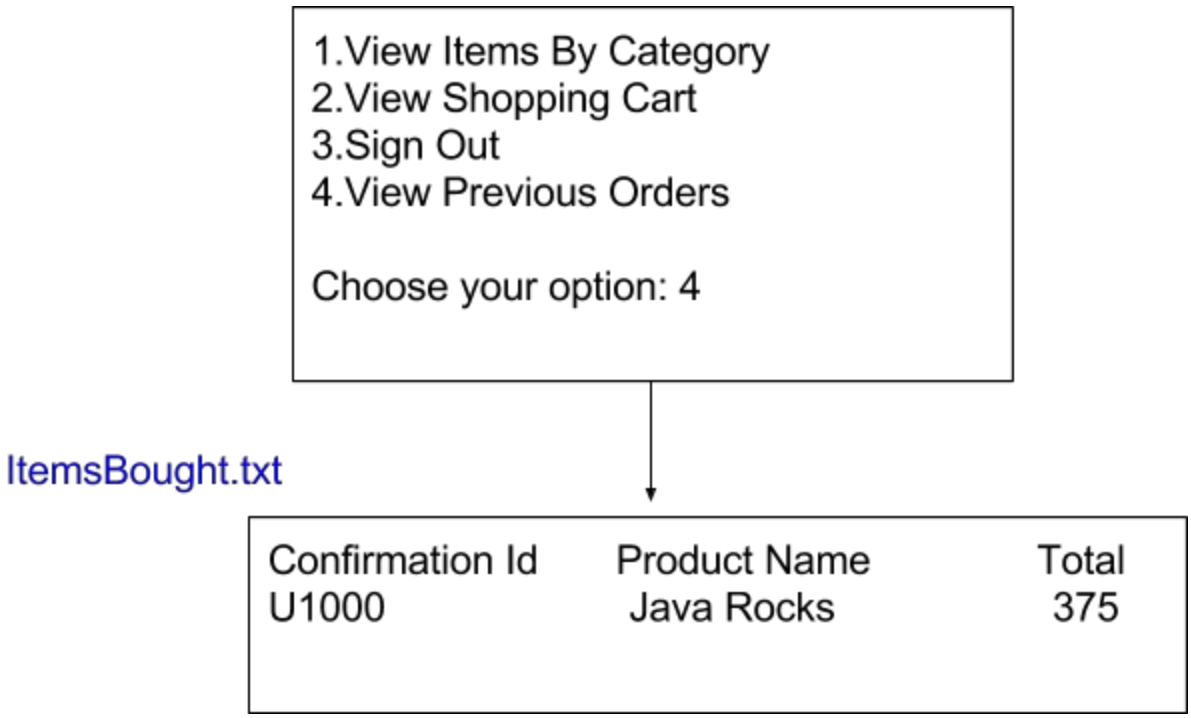
The following preamble **must** be written at the top of each .java source files:

```
/*
 * Name: [Full name of (no nicknames or chosen names) of each group member]
 * MacID: [MacIDs of all group members]
 * Student Number: [Student number of all group members]
 * Description: [This is an informative excerpt about this file.]
 */
```

Failure in meeting this format and file naming convention will result in 0 credit.

Extra credit

1. **(+2)** All details of the items purchased is stored in ItemsBought.txt. Details include Confirmation No, product name, and total amount spent.
   Also in the menu add a new item “View Previous Orders” as shown below.This must retrieve all information from ItemsBought.txt and display it on th**e screen**



2. **(+2)** Create a new User for **ADMIN** with ADMIN password. ADMIN must have the permission to sort the Items according to their price, names. ADMIN should also be allowed to add a new Item and also have a password which can be changed. ADMIN must not allow duplicate username in the Users.txt. The design of user interface for Admin is up to you.