create a new one if a saved table does not exist.

On startup, the `AuctionSystem` should check to see if the file `auctions.obj` exists in the current directory. If it does, then the file should be loaded and deserialized into an `AuctionTable` for new auctions/bids (*See the section below for information on the `Serializable` interface*). If the file does not exist, an empty `AuctionTable` object should be created and used instead. Next, the user should be promted to enter a username to access the system. This is the name that will be used to create new auctions and bid on the open auctions available in the table.

When the user enters 'Q' to quit the program, the acution table should be serialized to the file `auctions.obj`. That way, the next time the program is run, the auctions will remain in the database and allow different users to make bids on items. If you would like to 'reset' the auction table, simply delete the `auctions.obj` file.

- An `AuctionTable` member variable:
  - `auctionTable`
- A `String` member variable:
  - `username`
- `public static void main(String[] args)`
  - *Brief:*
    - The method should first prompt the user for a username. This should be stored in `username` The rest of the program will be executed on behalf of this user. Implement the following menu options:

- `(D) - Import Data from URL`
- `(A) - Create a New Auction`
- `(B) - Bid on an Item`
- `(I) - Get Info on Auction`
- `(P) - Print All Auctions`
- `(R) - Remove Expired Auctions`
- `(T) - Let Time Pass`
- `(Q) - Quit`

## Serializable Interface

You will also work with the idea of persistence. This means that our program should save all data from session to session. When we terminate a program, normally the data will be lost. We will preserve this data by using `Serializable` Java API and binary object files. All your classes should simply implement the java.io.Serializartion interface.

*Example:* Your `AuctionTable` class contains information for all auctions saved in the electronic database. You would want to preserve this data, so you can load this data the next time you run your program. You would do the following:

1. Modify the AuctionTable so that it implements the `Serializable` interface. Also, the `Auction` class should also make this implementation. No other changes are necessary.

```
public class AuctionTable implements Serializable
{
    // Member methods as is
}
```

2. In your application that contains the hash, you can include code that will save the hash into a file so it can be read in again later. To do this, you need to create an `ObjectOutputStream` to send the data to, and then use the `writeObject` method to send the hash to the stream, which is stored in the specified file.

```
FileOutputStream file = new FileOutputStream("auction.obj");
ObjectOutputStream outStream = new ObjectOutputStream(file);
AuctionTable auctions = new AuctionTable(/*Constructor Parameters*/);

// missing code here sets up the hash in some way

// the following line will save the object in the file
outStream.writeObject(auctions);
```

3. When the same application (or another application) runs again, you can initialize the member using the serialized data saved from step 2 so you don't have to recreate the object from scratch. To do this, you need to create an `ObjectInputStream` to read the data from, and then use the `readObject` method to read the hash from the stream.

```
FileInputStream file = new FileInputStream("auction.obj");
ObjectInputStream inStream = new ObjectInputStream(file);
AuctionTable auctions;

auctions = (AuctionLibrary) inStream.readObject();

// missing code here can use the same hash from step 2 now
```

## Sample Input/Output:

// Comment in green, input in red, output in black

```
Starting...
No previous auction table detected.
Creating new table...                               // Notify if loading or not.

Please select a username: student@stonybrook.edu

Menu:
    (D) - Import Data from URL
    (A) - Create a New Auction
    (B) - Bid on an Item
    (I) - Get Info on Auction
    (P) - Print All Auctions
    (R) - Remove Expired Auctions
    (T) - Let Time Pass
    (Q) - Quit


Please select an option: D
Please enter a URL: http://tinyurl.com/nbf5g2h
// TinyURL for http://www.cs.washington.edu/research/xmldatasets/data/auctions/ebay.xml

Loading...
Auction data loaded successfully!

// Menu not printed in sample i/o.

Please select an option: P

 Auction ID |      Bid    |        Seller          |          Buyer            |    Time    |  Item Info // truncated to
===================================================================================================================
  511601118 | $    620.00 | cubsfantony            |  gosha555@excite.com      | 110 hours  | Pentium III 933 System - 2
  511448507 | $    620.00 | ct-inc                 |  petitjc@yahoo.com        |  54 hours  | Pentium III 800EB-MHz Copp
  511443245 | $ 1,025.00  | ct-inc                 |  hsclm9@peganet.com       |  54 hours  | Intel Pentium III 933EB-MH
  511364992 | $    610.00 | bestbuys4systems       |  wizbang4                 |  55 hours  | Genuine Intel Pentium III
  511357667 | $    535.00 | sales@ctgcom.com       |  chul2@mail.utexas.edu    |  55 hours  | INTEL Pentium III 800MHz -


Please select an option: B
Please enter an Auction ID: 511364992

Auction 511364992 is OPEN
    Current Bid: $ 610.00

What would you like to bid?: 625.00
Bid accepted.

// Menu not printed in sample i/o.

Please select an option: P

 Auction ID |      Bid    |        Seller          |          Buyer            |    Time    |  Item Info // truncated to
===================================================================================================================
  511601118 | $    620.00 | cubsfantony            |  gosha555@excite.com      | 110 hours  | Pentium III 933 System - 2
  511448507 | $    620.00 | ct-inc                 |  petitjc@yahoo.com        |  54 hours  | Pentium III 800EB-MHz Copp
  511443245 | $ 1,025.00  | ct-inc                 |  hsclm9@peganet.com       |  54 hours  | Intel Pentium III 933EB-MH
  511364992 | $    625.00 | bestbuys4systems       |  student@stonybrook.edu   |  55 hours  | Genuine Intel Pentium III
  511357667 | $    535.00 | sales@ctgcom.com       |  chul2@mail.utexas.edu    |  55 hours  | INTEL Pentium III 800MHz -

// Menu not printed in sample i/o.

Please select an option: A

Creating new Auction as student@stonybrook.edu.
Please enter an Auction ID: 100000000
Please enter an Auction time (hours): 24
Please enter some Item Info: Core i5 2.7GHz - 4GB DDR3 - 750GB HDD

Auction 100000000 inserted into table.

// Menu not printed in sample i/o.

Please select an option: P

 Auction ID |      Bid    |        Seller          |          Buyer            |    Time    |  Item Info // truncated to
===================================================================================================================
  511601118 | $    620.00 | cubsfantony            |  gosha555@excite.com      | 110 hours  | Pentium III 933 System - 2
  511448507 | $    620.00 | ct-inc                 |  petitjc@yahoo.com        |  54 hours  | Pentium III 800EB-MHz Copp
  511443245 | $ 1,025.00  | ct-inc                 |  hsclm9@peganet.com       |  54 hours  | Intel Pentium III 933EB-MH
  511364992 | $    625.00 | bestbuys4systems       |  student@stonybrook.edu   |  55 hours  | Genuine Intel Pentium III
  511357667 | $    535.00 | sales@ctgcom.com       |  chul2@mail.utexas.edu    |  55 hours  | INTEL Pentium III 800MHz -
  100000000 |             | student@stonybrook.edu|                            |  24 hours  | Core i5 2.7GHz - 4GB DDR3
```

```
// Note that Bid and Buyer are missing for the new record. These correspond to 'null' attributes in the database.

// Menu not printed in sample i/o.

Please select an option: Q

Writing Auction Table to file...                // Serialize the Auction Table.
Done!

Goodbye.

// Starting up again.
Starting...
Loading previous Auction Table...               // Loading from file.

Please select a username: buyer@gmail.com

// Menu not printed in sample i/o.

Please select an option: B
Please enter an Auction ID: 100000000

Auction 100000000 is OPEN
    Current Bid: None

What would you like to bid?: 1500.00
Bid accepted.

// Menu not printed in sample i/o.

Please select an option: P

 Auction ID |      Bid   |        Seller         |          Buyer          |    Time   |  Item Info // truncated to
===================================================================================================================
  511601118 | $   620.00 | cubsfantony           |  gosha555@excite.com    | 110 hours | Pentium III 933 System - 2
  511448507 | $   620.00 | ct-inc                |  petitjc@yahoo.com      |  54 hours | Pentium III 800EB-MHz Copp
  511443245 | $ 1,025.00 | ct-inc                |  hsclm9@peganet.com     |  54 hours | Intel Pentium III 933EB-MH
  511364992 | $   625.00 | bestbuys4systems      |  student@stonybrook.edu |  55 hours | Genuine Intel Pentium III
  511357667 | $   535.00 | sales@ctgcom.com      |  chul2@mail.utexas.edu  |  55 hours | INTEL Pentium III 800MHz -
  100000000 | $ 1,500.00 | student@stonybrook.edu|  buyer@gmail.com        |  24 hours | Core i5 2.7GHz - 4GB DDR3

// Menu not printed in sample i/o.

Please select an option: T
How many hours should pass: 30

Time passing...
Auction times updated.

// Menu not printed in sample i/o.

Please select an option: P

 Auction ID |      Bid   |        Seller         |          Buyer          |    Time   |  Item Info // truncated to
===================================================================================================================
  511601118 | $   620.00 | cubsfantony           |  gosha555@excite.com    |  80 hours | Pentium III 933 System - 2
  511448507 | $   620.00 | ct-inc                |  petitjc@yahoo.com      |  24 hours | Pentium III 800EB-MHz Copp
  511443245 | $ 1,025.00 | ct-inc                |  hsclm9@peganet.com     |  24 hours | Intel Pentium III 933EB-MH
  511364992 | $   625.00 | bestbuys4systems      |  student@stonybrook.edu |  25 hours | Genuine Intel Pentium III
  511357667 | $   535.00 | sales@ctgcom.com      |  chul2@mail.utexas.edu  |  25 hours | INTEL Pentium III 800MHz -
  100000000 | $ 1,500.00 | student@stonybrook.edu|  buyer@gmail.com        |   0 hours | Core i5 2.7GHz - 4GB DDR3

// Menu not printed in sample i/o.

Please select an option: B
Please enter an Auction ID: 100000000

Auction 100000000 is CLOSED
    Current Bid: $ 1,500.00

You can no longer bid on this item.

// Menu not printed in sample i/o.

Please select an option: I
Please enter an Auction ID: 511448507

Auction 51148507:
    Seller: ct-inc
    Buyer: petitjc@yahoo.com
    Time: 24 hours
    Info: Pentium III 800EB-MHz Coppermine CPU - 256MB PC133 SDRAM - 30.7GB IBM Deskstar ATA100 7200RPM

// Menu not printed in sample i/o.

Please select an option: R

Removing expired auctions...
All expired auctions removed.

// Menu not printed in sample i/o.

Please select an option: P

 Auction ID |      Bid    |        Seller           |          Buyer           |    Time   |  Item Info // truncated to
===================================================================================================================
  511601118 | $   620.00 | cubsfantony             |  gosha555@excite.com    |  80 hours | Pentium III 933 System - 2
  511448507 | $   620.00 | ct-inc                  |  petitjc@yahoo.com      |  24 hours | Pentium III 800EB-MHz Copp
  511443245 | $ 1,025.00 | ct-inc                  |  hsclm9@peganet.com     |  24 hours | Intel Pentium III 933EB-MH
  511364992 | $   625.00 | bestbuys4systems        |  student@stonybrook.edu |  25 hours | Genuine Intel Pentium III
  511357667 | $   535.00 | sales@ctgcom.com        |  chul2@mail.utexas.edu  |  25 hours | INTEL Pentium III 800MHz -

// Menu not printed in sample i/o.

Please select an option: Q

Writing Auction Table to file...                // Serialize the Auction Table.
Done!

Goodbye.
```

Course Info | Schedule | Sections | Announcements | Homework | Exams | Help/FAQ | Grades | HOME