

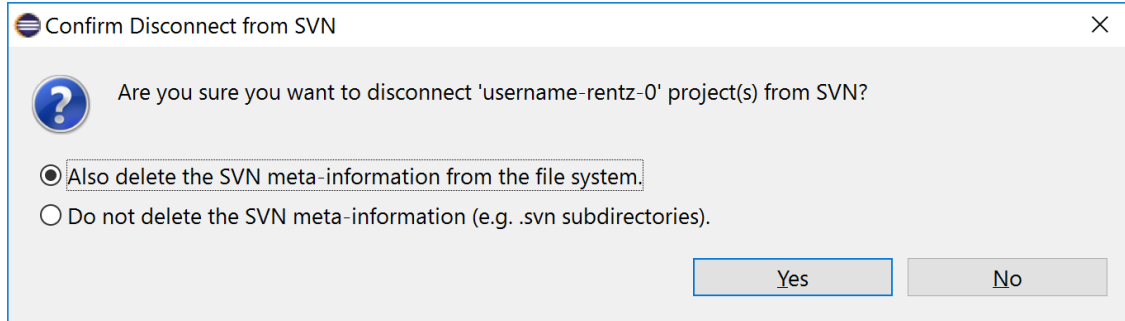
TCSS 305 Programming Practicum
Assignment 1: Login/Registration for Rentz
Value: 10% of the course grade
Due: Sunday, 20 October 2019, 23:59:00

Program Description:

Assignment 0, Assignment 1, 2, 3, 4 aim to develop Rentz, a vehicle rental system. You will incrementally build your code for each assignment, i.e., your Assignment 1 will be built using the code that you submitted for Assignment 0. Make sure you finish every assignment on time, so that your future submissions remain unaffected. Assignment 1 focuses on developing a Registration and login system for Rentz.

Using Previous Code:

1. You have already submitted the project “username-rentz-0” which is present in your project explorer to the SVN repository. Since you are going to build on the same code, you will need to rename your project. However, you will need to disconnect from the SVN repository before you rename your project.
2. Right-click on the project name on Package Explorer (in this case, “username-rentz-0”) and select “Team / Disconnect”. There will be a dialog box, select “Also delete the SVN meta-information from the file system” option and click “yes”. The project will be disconnected from SVN repository.



3. Save the contents of your project to a local drive. Right-click on the project name, select `Export -> Archive file`, then click “Next>”. In “To Archive File” enter the location and name of the archive that you want to create and click “Finish”. Your project will be saved in the given location.
4. To rename the project, right-click on the project name and select “Refactor->Rename”. You should rename the project to “username-rentz-1”, where “username” is your UWNetID. Once you have re-named the project, you can make changes to the code as needed. To submit the new code, i.e., “username-rentz-1” to SVN follow steps 1-6 under “Submitting your project to the SVN server” given in “hw0.pdf”. While following steps 1-6 from the document, replace “username-rentz-0” by the new project name “username-rentz-1”.

Implementation Guidelines:

NOTE: Do not change the name of the class fields and the methods as well as the method signatures. Do not add any extra fields other than `static final` constants. You can have different names for method parameters. You can define extra utility methods but should not remove any of the existing code. You should add appropriate access modifiers for the class members. Portions of the code have already been given to you. Your task is to complete the implementation of the `User` and `Registration` classes to make the code work.

`User`: this class represents information about the registered user. The outline of the class is already present in `model/User.java`

Field	Data Type	Description
<code>myName</code>	<code>String</code>	It represents the username
<code>myPassword</code>	<code>String</code>	It represents the password for login
<code>myVIPStatus</code>	<code>boolean</code>	It represents if a User is a VIP member or not

Method	Parameters	Return Type	Description
Parameterized Constructor	<code>String theName</code> <code>String thePassword</code>		Constructor with username and password. This constructor is called when a <code>User</code> object is created only with username and password. Default value for <code>myVIPStatus = false</code> should be assigned. The constructor should throw <code>NullPointerException</code> for null parameters and should throw <code>IllegalArgumentException</code> if the username or password is an empty string
Parameterized Constructor	<code>String theName</code> <code>String thePassword</code> <code>boolean theVIPStatus</code>		Constructor with username, password and <code>VIPStatus</code> The constructor should throw <code>NullPointerException</code> for null parameters and should throw <code>IllegalArgumentException</code> if the username or password is an empty string
<code>getMyName</code>		<code>String</code>	Returns the username
<code>getMyPassword</code>		<code>String</code>	Returns the password
<code>getMyVIPStatus</code>		<code>boolean</code>	Returns the <code>VIP Status</code>
<code>toString</code>		<code>String</code>	Returns the <code>String</code> representation of the <code>User</code> Object. The format of the string should be <code>classname (username, password, VIPStatus)</code> e.g. <code>User (TCSS305, 305, true)</code>
<code>equals</code>	<code>Object theOtherObject</code>	<code>boolean</code>	Checks if <code>theOtherObject</code> is equal to this object based on the username, password, <code>VIPStatus</code> fields
<code>hashCode</code>		<code>int</code>	Creates the hascode using username, password and <code>VIPStatus</code> fields

Registration: this class displays options for registration or login. It adds a new user to the list of registered users and saves it to file. It also verifies if the username and password of an existing user is correct during login.

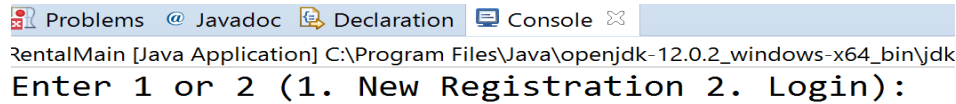
Field	Data Type	Description
USERFILE_NAME	static final String	It represents the file which saves the list of registered users consisting of user name, password, VIPstatus for each of them. Do not change the value of this field. The file is present under resources/registeredusers.txt
myUserList	final Map<String, User>	HashMap representing the list of registered users, where key is the username and value is the User object

Method	Parameters	Return Type	Description
Parameterless Constructor			Loads the list of previously registered user from the file.
printSignin		void	<p>Prints user options in the console output. Displays option for new registration or login.</p> <p>For new user registration, it accepts user name, password and VIPStatus from console. It calls the <code>register</code> function to verify if registration is successful or one needs to re-enter the username, password and VIPStatus.</p> <p>For user login, it checks accepts username and password from console. It calls <code>login</code> method to check if the credentials are correct or if they need to be re-entered.</p> <p>The exact format of the output that is required for implementing the <code>printSignin</code> function is shown using screenshots below.</p>
login	String theUserName String thePassword	boolean	<p>Checks if the username is present in the list of registered users. If the user is already present and if the password matches the entered password, it returns <code>true</code>. Otherwise, it returns <code>false</code>.</p> <p>The parameters should be checked for null values and should throw <code>NullPointerException</code> for null parameters</p>
register	User theUser	boolean	<p>Adds <code>theUser</code> to <code>myUserList</code></p> <p>Write the registered user to file. The information of every registered user should be “added” to the file <code>resources/registeredusers.txt</code>. To write to the file use the <code>writeUserToFile(final String theFile, final User theUser)</code> method from the <code>Fileloader</code> class present in the utility package</p>
clear		void	Clears and removes all elements in <code>myUserList</code>
toString		String	Returns the String representation of the Registration Object. The format of the string should be Registered UserList {key1=value1,key2=value2}

			e.g., Registered UserList {tcss305=User (tcss305, 305, true), athirai= User (athirai, 1234, true)}
--	--	--	--

Your program should display the exact output as shown below

1. When you run RentalMain.java you should get the following output

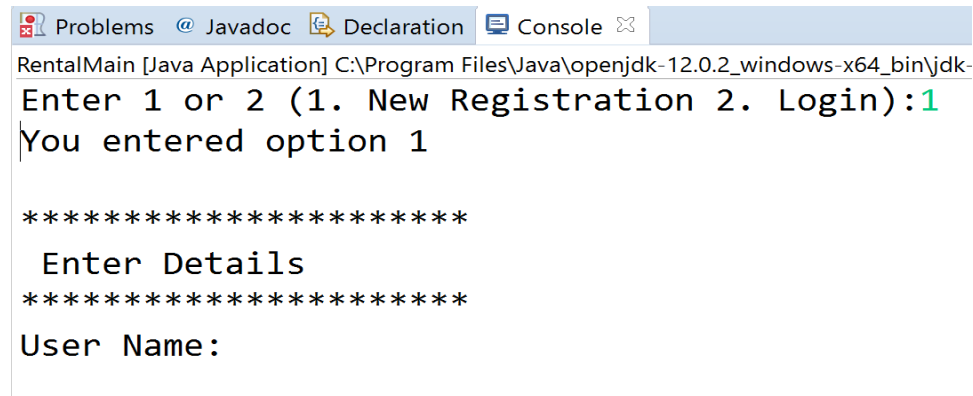


```

Problems @ Javadoc Declaration Console
RentalMain [Java Application] C:\Program Files\Java\openjdk-12.0.2_windows-x64_bin\jdk-
Enter 1 or 2 (1. New Registration 2. Login):

```

2. When 1 is entered, the following should be displayed



```

Problems @ Javadoc Declaration Console
RentalMain [Java Application] C:\Program Files\Java\openjdk-12.0.2_windows-x64_bin\jdk-
Enter 1 or 2 (1. New Registration 2. Login):1
You entered option 1

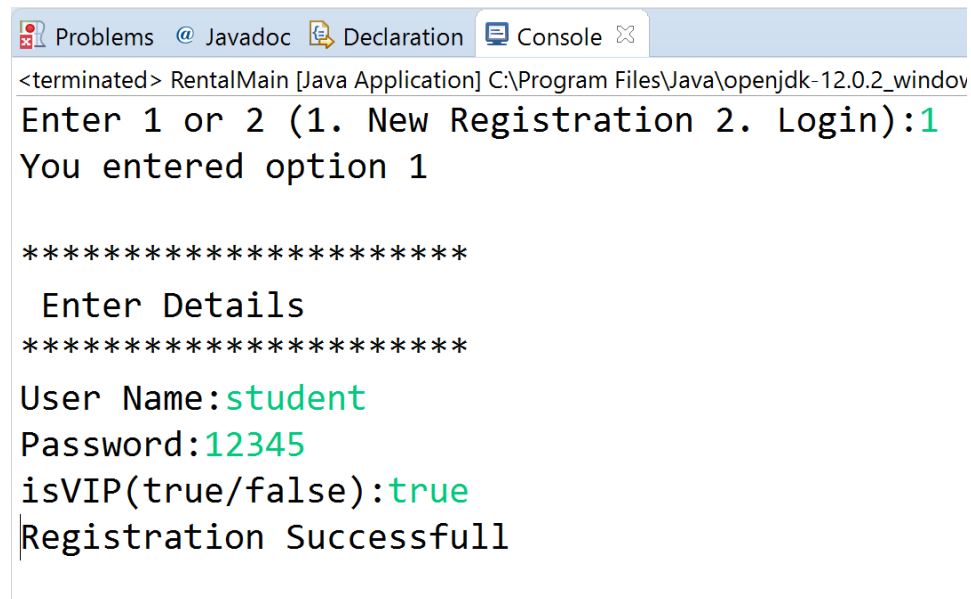
*****

Enter Details
*****

User Name:

```

3. After entering the user details for new registration, the following should be displayed If it is successful



```

Problems @ Javadoc Declaration Console
<terminated> RentalMain [Java Application] C:\Program Files\Java\openjdk-12.0.2_windows-x64_bin\jdk-
Enter 1 or 2 (1. New Registration 2. Login):1
You entered option 1

*****

Enter Details
*****

User Name:student
Password:12345
isVIP(true/false):true
Registration Successfull

```

4. If the username already exists (even though password, VIPStatus are different), the system asks for a new username and password until a non-existing username and any password is entered.

```
Problems @ Javadoc Declaration Console
<terminated> RentalMain [Java Application] C:\Program Files\Java\openjdk-12.0.2_windows-x64_bin\jdk-12.0.2
Enter 1 or 2 (1. New Registration 2. Login):1
You entered option 1

*****

Enter Details
*****

User Name:athirai
Password:123456
User already exists, enter different user name:athirai
Enter Password:1234
User already exists, enter different user name:athirai
Enter Password:43567
User already exists, enter different user name:student305
Enter Password:32456
isVIP(true/false):true
Registration Successfull
```

5. When option 2 is selected, the following output should be displayed

```
Problems @ Javadoc Declaration Console
RentalMain [Java Application] C:\Program Files\Java\openjdk-12.0.2_windows-x64_bin\jdk-
Enter 1 or 2 (1. New Registration 2. Login):2
You entered option 2

*****

Enter Details
*****

User Name:
```

6. When the username and password is entered correctly, the following should be displayed

```
Problems @ Javadoc Declaration Console
<terminated> RentalMain [Java Application] C:\Program Files\Java\openjdk-12.0.2_windo
Enter 1 or 2 (1. New Registration 2. Login):2
You entered option 2

*****

Enter Details
*****

User Name:athirai
Password:1234
Login Successfull
```

7. When the username or password is entered in-correctly, the username and password should be re-entered until the credentials are found to be correct

```
Problems @ Javadoc Declaration Console
<terminated> RentalMain [Java Application] C:\Program Files\Java\openjdk-12.0.2_windo
Enter 1 or 2 (1. New Registration 2. Login):2
You entered option 2

*****

Enter Details
*****

User Name:305student
Password:1234

Wrong Credentials
Enter User Name:athirai
Enter Password4568

Wrong Credentials
Enter User Name:athirai
Enter Password1234
Login Successfull
```

Extra Credit: Up to 5% extra credit can be earned by completing the following. The password does not have any restrictions currently. If you want the user to enter password with restrictions Step 7 should display “Password Does not Comply [...List the rules the password should follow]”, when a password that does not comply with your rules is entered. The user should be allowed to re-enter a password until it complies with the rules. You should enforce at least 5 rules of your own choice.

Submitting your executive summary:

The executive summary template will be provided to you in canvas. Download the file. Rename the executive summary “executivesummary-username-rentz-1”, where username is your UWNetID. Fill and upload the *executive summary* to canvas as the same WORD format. The grader will grade your submission based on the executive summary as well as the project you submit to the SVN repository. If you do not submit either of them, you will receive no grades.

Grading Rubric (Total 100 points, 105 possible with extra credit):

Task	Max Score Possible
Executive Summary -Submission on canvas with correct version number	5
Source Code – Submission to SVN	5
User class – correct fields as per guidelines	5
User class – correct access modifiers for fields and methods	5
User class – correct implementation of parameterized constructor1	5
User class – correct implementation of parameterized constructor2	5
User class – correct implementation of getName(), getMyPassword(), get MyVIPStatus()	5
User class – correct implementation of toString method	5
User class – correct implementation of equals method	5
User class – correct implementation of hashCode method	5
Registration class – correct implementation of toString() and clear()	5
Registration class –printSignin: display New Registration/Login option [code: 3, output: 2]	5
Registration class – printSignin (new registration): get user input for Username, password, VIPStatus [code: 3, output: 2]	5
Registration class – printSignin, register: check if user already exists and print login successful [code: 3, output: 2];	5
Registration class – printSignin, register: check if user already exists and print User already exists until a correct username is entered [code: 3, output: 2]	5
Registration class –printSignin, login: get username and password and call login function to check if successful [code: 3, output: 2]	5
Registration class –printSignin, login: get username and password and call login function until an existing username and correct password is entered else display wrong credentials [code: 3, output: 2]	5
Correct Project Name and Executive summary name	5
No Errors and Warnings (except reasonable ones which should be clearly explained in executive summary)	5
Proper Java docs and Header comments	5
Extra Credit	5
Total (105 points possible with extra credit)	100