

Assigned: September 14, 2010

Due: Sunday, September 18, 2010 by 11:59 PM

Deliverables:

The following project files must be uploaded to Web-CAT by the due date and time specified above (see the Lab Guidelines for information on submitting project files).

- UserID.java
- GenerateUserID.java

Directions:

UserID.java

- **Requirements:** You have been requested to create a program that allows a user to enter their first name and last name and generate a random user id and default password (a 6-digit random number). The user id must contain the first 3 letters of the user's last name, the first three letters of the user's first name, and a random number. The random number must have the following characteristics:
 - The number must be from 000 to 199
 - The second digit must be from 3-9
 - The third digit must be from 0-9

Example:

First name: Bob

Last name: Smith

Example user name: smibob143

Example random password: 386425

- **Design:** Your UserID class must contain a constructor that accepts 2 strings as parameters representing the user's first name and last name. A user id and password must also be generated. Example:

```
UserID myInfo = new UserID("Bob", "Smith");
```

The UserID class must also contain the following methods:

- `getId`: accepts no parameters and returns the user name that has been generated for the user as a String. The same user name must be returned every time the method is called once the object has been created (HINT: you will want the user name to be generated for each object when it is created rather than when the `getId` method is invoked). If the user's first or last name is less than 3 characters long, then concatenate the existing characters with the random number (Example: Jane Ki might be janki037).

- `getPassword`: accepts no parameters and returns a `String` representing the current password. The current password must be randomly generated when the object is created and only changed when `generateNewPassword` is invoked or when the password is set using the `setPassword` method.
 - `setPassword`: takes a `String` parameter and returns a `boolean`. If the string parameter (the new password) is less than 6 characters, then the method returns `false` and the password is not set. Otherwise, the password is set to the parameter value and the method returns `true`.
 - `generateNewPassword`: Takes no parameters and does not return a value. The method should generate a new password for the user. The password must be a 6-digit number.
 - `toString`: Returns a `String` containing the user's first name and last name on the first line, their user name on the second line, and their current password on the third line. Example:
Bob Smith
smibob143
386425
- **Code**: Note that the user ID must contain a number from 000 to 199, but the second digit must not be below 3. You can therefore generate integers in the 3-digit number separately and then concatenate them together. Use constant fields instead of including literal values in your code. Also, keep variables local where possible.
 - **Test**: Make sure that once an object is created, the user ID always stays the same. The password should also be generated when the object is created, and only changed if `setPassword` is invoked with a valid input or if `generateNewPassword` is invoked.

Directions:**GenerateUserID.java**

- **Requirements**: The `GenerateUserID` program accepts a user's first name and last name as input, prints out the user's information as shown in the `toString` method of `UserID`, prompts the user to change the password, (if yes) asks for a password, and then prints out the user's information again as shown in the `toString` method of `UserID`.
- **Design**: Your `GenerateUserID` class is a driver program, meaning that the class should contain a main method. Follow the output of the examples on the next page. Replace everything in italics with your own words; orange text is example user input.

Example: Replace password; valid password entered.

Line #	Program output
1	<i>Prompt user for first name: Jane</i>
2	<i>Prompt user for last name: Winters</i>
3	
4	Name: Jane Winters
5	User ID: winjan062
6	Password: 245664
7	
8	<i>Would you like to change your password (y - Yes, n - No) Y</i>
9	
10	<i>Prompt user for password: awdkua37</i>
11	
12	Name: Jane Winters
13	User ID: winjan062
14	Password: awdkua

Example: Replace password; invalid password entered.

Line #	Program output
1	<i>Prompt user for first name: Jane</i>
2	<i>Prompt user for last name: Winters</i>
3	
4	Name: Jane Winters
5	User ID: winjan170
6	Password: 19459
7	
8	<i>Would you like to change your password (y - Yes, n - No) Y</i>
9	
10	<i>Prompt user for password: awdku</i>
11	
12	Error: Invalid password. Password must be 6 or more digits.

Example: Do not replace password.

Line #	Program output
1	<i>Prompt user for first name: Jane</i>
2	<i>Prompt user for last name: Winters</i>
3	
4	Name: Jane Winters
5	User ID: winjan189
6	Password: 87734
7	
8	<i>Would you like to change your password (y - Yes, n - No) n</i>

- **Code & Test:** User's should be able to enter y or Y for yes and n or N for no. Do not repeat code from the UserID class (use the boolean return from setPassword to determine whether the password is valid). Example (idGen is of type UserID and password is of type String):

```
if (idGen.setPassword(password)) // when id is valid
```