

CS161: Introduction to Computer Science
Homework Assignment 3
due 2/7 by 11:59pm

Scanner, String, and Random

The purpose of this homework assignment is to give you practice creating and using objects of type `Scanner`, `String`, and `Random`. Recall that these Java classes were written by some other programmer for your use.

Creating objects in Java requires you to use the `new` keyword. Once an object is created, you can call (or invoke) a method on the object by using the dot operator. The most important thing to remember when creating and using objects is that all of your *types* need to match!

Written Exercises

Double click on the sheet of notebook paper icon in BlueJ. This will open a README file where you can type your answers to the following exercises:

- Complete exercises 3.1, 3.2, 3.5, and 3.7 in the textbook
- Consider the following Java program (you can ignore the line numbers for now):

```
1 /**
2  * This class simulates the roll of an n-sided die.
3  * @author alchambers
4  */
5 public class RollDie{
6     public static void main(String[] args){
7         Random rand = new Random();
8         Scanner input = new Scanner(System.in);
9
10        System.out.println("Enter the number of sides of the die: ");
11        int sides = input.nextInt();
12
13        // generate a random int between [1, sides]
14        int roll = rand.nextInt(sides)+1;
15        System.out.println("Roll: " + roll);
16    }
17 }
```

Answer the following questions using the code above.

1. What is the type of the variable `rand`?
2. What is the type of the variable `input`?
3. On line 11, the method `nextInt()` is called. What class contains this method?
4. On line 14, another method named `nextInt()` is called. What class contains this method?

Generating Usernames and Passwords

Now that we know about the `Scanner` class, we can finally write interactive programs! This week we'll mix some input with some string processing, toss in a bit of randomness, and produce a program that generates faux user names and suggested passwords for users.

Your program should prompt the user for three pieces of information: their first name, last name, and favorite word. You'll then use those pieces of information to help generate something that could be used as a unique user name, as well as two suggested passwords, each constructed using different rules. A sample interaction is shown below:

```
Welcome to the username and password generator!
Please enter your first name: Ron
Please enter your last name: Thomas
Please enter your favorite word: Literature

Thanks Ron, your user name is rthomas*84

Here are two possible passwords:

Password 1: r0n39th0m@$
Password 2: RonLiteraTh
```

Programming Questions

1. Create a BlueJ project called `hw3` and inside create a Java class named `Passwords`. All of your code should be in the `main` method of this class.
2. Add code that uses a `Scanner` to read the user's first name, last name, and favorite word.
3. The username generated by your program should consist of:
 - the first letter from the user's first name
 - followed by the first seven letters from their last name
 - followed by a random integer between 0 and 99

The letters in the username should all be lower case. You should add `*` characters as necessary if the last name is shorter than seven characters. (Hint: Add some extra `*`'s to the last name before you select the seven-character piece, whether you need them or not.)

For full credit, *your solution should build a single string containing all of these characters and then print it, rather than just printing each piece separately.* You should also be polite and personalize the response by including the user's first name, as shown above.

4. The first password should consist of:
 - the user's first name
 - a random integer between 0 and 99
 - the user's last name

Again, the letters in the password should all be lower case. Once you construct the password, some of the characters can then be replaced by similar looking digits and punctuation characters. You should perform the following replacements, though you can feel free to add some more of your own: `'a'` by `'@'`, `'o'` by `'0'`, `'l'` by `'1'`, and `'s'` by `'$'`.

5. The second password consists of:

- a random-length portion of the first name
- a random-length portion of the favorite word
- a random-length portion of the last name

In each case, those random-length pieces should start at the beginning of the string, and the code should be written such that it's possible to get the entire string if the largest possible random number is produced.

6. Before submitting your assignment, make sure to comment your code. There should be a Javadoc comment at the top of the class containing a brief description, your name, and the date. In addition, you should add inline comments above each major section of your code – e.g., comments describing how the username-creation code works, and comments for each of the password approaches.

Extras

Looking for additional challenges? Feel free to invent some additional username or password formats and implement them in addition to the required ones.

For example, instead of taking random-length portions of the inputs that always start at the beginning of the input strings, take substrings where both the starting and ending points are random. You could also ask for an additional input (e.g. the user's favorite number) and use that in interesting ways.

Submitting your assignment

You should submit your `hw3` folder with all of your work inside. Please remember to rename your folder

`hw3_firstName.lastName`

before you zip it.