# **CSC-151 Midterm Exam Spring 2021, Part 2**

## **Objectives**

By the end of this program, the student will demonstrate the ability to

* Write static methods
* Use methods in a different class
* Use loops
* Use if/else statements
* Use single-dimension arrays
* Manipulate Strings

## **Password Generator**

Generate a list of unique passwords.  Your program should prompt the user for how many passwords they want to generate, then print the list of passwords. All passwords should follow this pattern:

1. Begin with a random four-letter word.  The starter code includes a static method to return a list of valid words.
2. One special character:  ! @ # \_ . \* The starter code includes a static method to return an array of the special characters.
3. A two-digit number: 00-99
4. Five random alphabetic or numeric characters from:  abcdefghijklmnopqrstuvwyxzABCDEFGHIJKLMNOPQRSTUVEWXYZ0123456789. The starter code includes a method to return these characters as a String

The password **rain\_23A6vW9** is an example that meets these rules.  Your program must include code that checks to see that all passwords are unique.

Complete the Java program PasswordGenerator, which is in the attached zip file. Implement the following methods:

### **main**

Complete the code in the main method so that the program prompts for the number of passwords, then creates and prints the specified number of unique passwords. Refer to the Sample Output below.

### **createPassword**

Complete the method **public** **static String createPassword()**. This method should return a password that meets the rules outlined above. Note that the starter code contains static methods to create four-letter words, special characters, and alphabetic/numeric characters.

### **isPasswordUnique**

Write the method **isPasswordUnique()**. This method should take 2 parameters: the password to check and a String containing all unique passwords created so far. Return true if the password is unique.

### **addUniquePassword**

Complete the method **public static void addUniquePassword(String aPassword, String passwordList)**. This method will return an updated String containing all unique passwords including aPassword.

## **Notes:**

* **Hint:** The starter code contains **140** unique four-letter words
* **Hint**: You can track the already-generated passwords in a String. Each time you create a new password, you can update the String of already-generated passwords
* Items in the program that you must complete have a TODO comment
  + EX: // TODO: Complete this method
* Assume that the user only enters valid input
* Complete this program in baby steps. There are many ways to approach this program, but whichever order you use, write something then test it immediately. For example, write one method and test it. Then write another and test it. And so on. It is better to have something that works, though it’s incomplete, than to have something complete that has nothing working.

## **Grading Elements:**

* Generated passwords meet the specified rules
* Generated passwords use the four-letter words, special characters, and password characters static methods provided in the starter code
* All passwords are unique
* Specified number of passwords are created
* Previously generated passwords are tracked efficiently
* Provided method signatures are not changed
* Any new methods have appropriate signatures

## **Sample Output**

How many passwords do you want to generate? **10**

1: tide\*19gH7pd

2: five.57SxJCk

3: rung@65NBTah

4: rote@310W4Yu

5: tide.15RdJKp

6: vile@56tU1Fd

7: five\*788BW37

8: rang#24baGa9

9: idle!835MJT6

10: flip@32bICLg