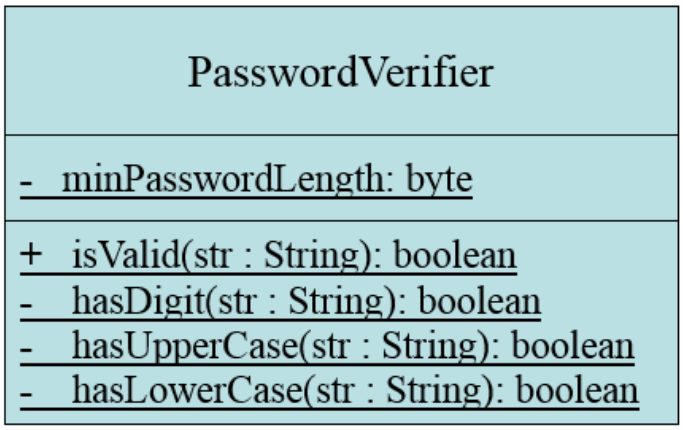
CS 111, Programming Fundamentals II
Lab 3: Password Verifier

Computer Science

This lab is meant to give you practice with static methods and Strings.

I. Password Verifier

Write the class **PasswordVerifier** according to the UML below. The methods ***hasDigit(), hasUpperCase()*** and ***hasLowerCase()*** are very similar. Once you write the first one you can copy/paste it and modify it slightly. The method ***isValid()*** is static and public, you will call this method when you need to validate the password. The password has to have an upper case character, a lower case character, a number and also has to be at **least 7 characters long**. You will check if it satisfies all of those requirements and if it does not, you will print to the screen which requirements are missing.



The password will be taken in as users input in a form of a ***String***. Remember that a string is essentially a ***char*** array, and we are able to index the characters inside of it using the method ***charAt***(). This method returns a ***char***, once we have a ***char*** variable we can use the wrapper class **Character,** which has methods that can check if the character is upper/lower case or if it's a digit. Below is an example of how to get a character out of a ***String*** at location 0 and check if it is upper case. Refer **Character** class in the Java API for more information and for the other two methods to check if it's lower case or if it is a digit.

```
String str = "Cwu";
char c = str.charAt(0);
boolean isUpper = Character.isUpperCase(c);
```

Inside of the ***isValid()*** method check if the password has enough characters. Then call the other three methods to check if it has met all of the requirements. Return true if it has satisfied all of the requirements, else return false. Along the way print out which of the requirements are not met as you are checking them. It is up to you how you structure the method calls inside of the ***isValid()*** method. Since this method is public and static, it can be called directly on the class name.

Inside if the method ***hasUpperCase()*** go through the String ***str*** using a loop**.** ***str*** is the parameter getting passed into the method. Inside of the loop get a character from the string and check it if it is upper case using the wrapper class, similar to the code snippet above. Return a true if the string contains an upper case character, else return false. Do the same for the methods ***hasLowerCase()*** and ***hasDigit(),*** simply change the method which checks the character.

Write another class named **MakePassword.java** which will prompt the user to enter a password. Check the password using the method ***isValid()*** in the **PasswordVerifier** class and print out the result. Sample outputs can be seen below in **Figure 1**.

```
 ----jGRASP exec: java MakePassword
Enter a password: cwu1D
Password needs to be atleast 7 characters long.
INVALID PASSWORD

 ----jGRASP: operation complete.

 ----jGRASP exec: java MakePassword
Enter a password: cwuwildcats
Password did not include a digit.
Password did not include an upper case character.
INVALID PASSWORD

 ----jGRASP: operation complete.

 ----jGRASP exec: java MakePassword
Enter a password: Wildcats
Password did not include a digit.
INVALID PASSWORD

 ----jGRASP: operation complete.

 ----jGRASP exec: java MakePassword
Enter a password: cwuCS2020
VALID PASSWORD

 ----jGRASP: operation complete.
```

Figure 1: Sample Outputs

II. Upload your work to Canvas

For this lab, make sure that you upload the following files to the Lab 3 assignment in your Canvas account:

PasswordVerifier.java
PasswordMaker.java
Screenshot image containing an output of the PasswordMaker

Rubric

File / Lab	Points
Class *PasswordVerifier* written per UML diagram and compiles	60
Class *PasswordMaker* prompts the user and uses the class *PasswordVerifier*	20
Class *PasswordMaker* runs as expected	10
Screenshot of the output, code is commented and formatted	10
Total	100