CS 1400-03 Introduction to Programming and Problem Solving
Coding Practice #10
(Due: 11:59 PM, Friday, 4/23/2021)

Except Coding Practice #1, I will not grade your coding practice submissions. Instead, they will be treated as participation points. On blackboard, you will receive full points as long as you work on the exercises, which don't necessary mean they are all correct. Please check your own programs carefully and make sure they do generate the desired output.

Objectives:

- Be able to use `String` methods: `indexOf`, `substring`
- Be able to use utility methods of wrapper classes
- Be able to test and debug a program

Change your working directory to `cs1400/codingPractice` for this assignment.

Task #1 Word Replacement

Write a program, called `WordReplacement.java`, that reads in a line of text and then outputs that line of text with the first occurrence of "hate" changed to "love". Use a loop to allow the user to repeatedly enter another line of text and terminate the program if input is a blank line. Your program shall use only the following `String` methods: `indexOf(aString)`, `substring(start)`, and `substring(start, end)`. The following are sample interactions where the user's input is shown in bold.

```
fcsang@garrison ~/cs1400/codingPractice $ java WordReplacement
enter a line of text (blank line to stop):
I hate you.
I have rephrased that line to read:
I love you.

enter another line (blank line to stop):
I hate, hate, hate you.
I have rephrased that line to read:
I love, hate, hate you.

enter another line (blank line to stop):
I love you.
"hate" is not found.

enter another line (blank line to stop):
<enter>
fcsang@garrison ~/cs1400/codingPractice $
```

Now write another program, called `WordReplacementWithMethod.java`, that defines a `static` method name `replaceSubstring`. This method accepts three `String` objects as arguments. Let's call them `string1`, `string2`, and `string3`. It searches `string1` for the first occurrence of `string2`. When it finds `string2`, it replaces it with `string3` and returns the result. Otherwise, it returns a message saying that `string2` is not found. The following are sample interactions where the user's input is shown in bold.

```
fcsang@garrison ~/cs1400/codingPractice $ java WordReplacementWithMethod
enter a line of text (blank line to stop):
I love you.
enter a substring to be replaced: hate
enter the new substring: love
"hate" is not found.

enter another line (blank line to stop):
I hate you.
enter a substring to be replaced: hate
enter the new substring: love, love, love
I have rephrased that line to read:
I love, love, love you.
```

Task #2 Password Verifier

Image you are developing a software package for Amazon.com that requires users to enter their own passwords. Your software requires that users' passwords meet the following criteria:

- The password should be at least six characters long.
- The password should contain at least one uppercase and at least one lowercase letter.
- The password should have at least one digit.

Write a utility class `PasswordVerifier` that verifies that a password meets the stated criteria. Your program should use a constant for the minimum number of characters for each password. Demonstrate the class in a test driver `PasswordTest` that allows the user to enter a password and then displays a message indicating whether it is valid or not. If not, why. The following show sample interactions:

```
fcsang@garrison ~/cs1400/codingPractice $ java PasswordTest
Enter a password: Hello2021
Valid password.

fcsang@garrison ~/cs1400/codingPractice $ java PasswordTest
Enter a password: 12 Cats
Valid password.

fcsang@garrison ~/cs1400/codingPractice $ java PasswordTest
Enter a password: aBC45
- length must be at least 6
Invalid password.

fcsang@garrison ~/cs1400/codingPractice $ java PasswordTest
Enter a password: 123456
- need at least one uppercase letter
- need at least one lowercase letter
Invalid password.

fcsang@garrison ~/cs1400/codingPractice $ java PasswordTest
Enter a password: abc
- length must be at least 6
- need at least one uppercase letter
- need at least one digit
Invalid password.
```

```
fcsang@garrison ~/cs1400/codingPractice $ java PasswordTest
Enter a password: +-*/
- length must be at least 6
- need at least one uppercase letter
- need at least one lowercase letter
- need at least one digit
Invalid password.
```

Submission:
Generate a `script` file `practice10.txt` with appropriate time stamps and the following steps visible:

1) a `pwd` to show the current working directory
2) a `ls -l` to show in long format the files in your `cs1400/codingPractice` directory
3) display `WordReplacement.java`
4) compile `WordReplacement.java`
5) run `WordReplacement`
6) display `WordReplacementWithMethod.java`
7) compile `WordReplacementWithMethod.java`
8) run `WordReplacementWithMethod`
9) display both `PasswordVerifier.java` and `PasswordTest.java`
10) compile `PasswordTest.java`
11) run `PasswordTest`

Submit the script file `practice10.txt` on Bb, under the Coding Practice Folder, Practice #10 link.