CS 1302A – HW 1 (Spring 2016)
Total: 100 pts
Due: Wednesday Jan 27, 2016 11:59pm

This homework has three problems dealing with String/Character methods.

Eclipse Reminder

- You will create a Java Project in Eclipse with the name: *hw1_FLastname*
- You will have three packages: *prob1*, *prob2*, *prob3*

Problem 1

Write a class, *MessageParser* that contains a static method, *message*, which accepts a string. This string may contain the code '#' followed by a digit (*d*) between 1 and 9. Your method should return the substring composed of the *d* characters *after* the digit. If the code is not found then return a null string. Examples: "abc#4efghnop" → "efgh", "abc4efghnop" →null. Write a *main* that prompts the user for a string and displays the result of calling the method on the string.

Reminder on how to use Eclipse:	d. Choose: New, Class
	e. Change the *Package* to "prob1"
a. Open Eclipse in your workspace.	f. Supply the class *Name* "*MessageParser*"
b. Create a new java project named, *hw1_FLastname*	g. Select option to generate *main*.
c. Right-click the *src* folder	h. Choose: Finish
	i. Write your code!

Problem 2

A *Level 1* password must be at least 6 characters long and meet at least 2 of the following conditions.

- at least one lower case letter
- at least one upper case letter
- as least one digit

A *Level 2* password must be at least 6 characters long and meet all 3 of the conditions above.

Write a class, *PasswordChecker* with the following static methods.

a. `boolean isLevel1(String password)` – returns *true* if *password* is a Level 1 password and false otherwise.
b. `boolean isLevel2 (String password)` – returns *true* if *password* is a Level 2 password and false otherwise.
c. `int passwordLevel (String password)` – returns 1 if *password* is a *Level 1* password, 2 if it is a *Level 2* password, and 0 otherwise.

Hint: you should write helper methods for each of the (bulleted) conditions. Then, the required methods will be easy to write.

Write a *main* that prompts the user for a password and displays the result of calling each of the methods. Suppose the user enters: *UL3REF66*. The output should be formatted as shown below.

```
Proposed Password: UL3REF66
isLevel1(): true
isLevel2(): false
passwordLevel(): 1
```

Reminder on how to use Eclipse:	e. Select option to generate *main*.
a. Right-click the *src* folder	f. Choose: Finish
b. Choose: New, Class	g. Write your code!
c. Change the *Package* to "prob2"	
d. Supply the class *Name* "PasswordChecker"	

Problem 3

Write a class, *WordReverse* with the following static method:

a. reverseSentence – This method accepts a string which is composed of words separated by spaces. It returns a string with the entire sentence reversed. For example:

   "The red king has the ring" → "gnir eht sah gnik der ehT"

   Hint: This may sound more complicated than it is! It doesn't matter that there are "words separated by spaces." You can just treat it as a long string and simply reverse the characters. (Still need more hint? One approach is to loop through the string backwards, extracting each character as you go and building the new string)

Write a *main* that prompts the user for a string and displays the result in the following format:

```
Input String: The red king has the ring
reverseSentence (): gnir eht sah gnik der ehT
```

Reminder on how to use Eclipse:	e. Select option to generate *main*.
a. Right-click the *src* folder	f. Choose: Finish
b. Choose: New, Class	g. Write your code!
c. Change the *Package* to "prob3"	
d. Supply the class *Name* "*WordReverse*"	

Submission

Follow the directions in Lab 1 to zip the folder *hw1_FLastname*.
Make sure you zip the ENTIRE folder *hw1_FLastname*!! Submit your zip file to Blazeview by the due date. The name of your file should be: *hw1_FLastname.zip*.

Grading

I will RANDOMLY select one problem for grading. For example, if problem 2 is selected, I will only go to the package/folder *prob2* for grading. So make sure all your Java codes are in the correct packages/folders.

Additional Requirements:

1) No late submission will be accepted.
2) Please exactly follow the naming rules described above. You will be deducted 10 points for incorrect naming.
3) Write comments at the beginning of your Java source file(s) to indicate your name, student ID, "CS 1302-A Homework 1", and the due date.
4) Make sure that your programs are well-documented, readable, and user friendly. Make sure you document your programs by adding comments to your statements/blocks so that I can understand your code. You will get 0 to 10 points based on the helpfulness of your comments. You will be deducted 10 points for no comments.
5) It is your responsibility to make sure your programs can compile and run correctly. Please be aware that programs that do not compile may receive ZERO credit.
6) When grades are returned to you via BlazeView, you have 7 days to meet with the instructor to discuss on grade changes. After 7 days, the grades are written in stone and can't be changed after that point.