Electronic submission to eCampus due at 23:59, Wednesday, 11/16/2016
By electronically submitting this assignment to eCampus by logging in to your account, you are signing electronically on the following Aggie Honor Code: "On my honor, as an Aggie, I have neither given nor received any unauthorized aid on any portion of the academic work included in this assignment."

Note 1: This homework set is individual homework, not a team-based effort. Discussion of the concept is encouraged, but actual write-up of the solutions must be done individually.

Note 2: Turn in one yourLastName-yourFirstName-hw5.tar or yourLastName-yourFirstName-hw5.zip file on eCampus, nothing else. What to submit is detailed below.

Note 3: All Java code that you submit must compile without errors using javac of Java 8 (that is installed in the departmental servers, linux.cse.tamu.edu and compute.cse.tamu.edu). If your code does not compile, you will likely receive zero points for this assignment.

Note 4: Remember to put the head comment in your files, including your name, your UIN, and acknowledgements of any help received in doing this assignment. Again, remember the honor code.

Note 5: Problem 1 contains total of 50 points (each method implementation contains 10 points). The Problem 2 contains total of 50 points. MyDate Class implementation contains 15 points. The rest of the program contains 35 points.

Note 6: Use the Scanner class in your test class to make your code more user-friendly. It is upto you how you will incorporate this into your program. Also, you can use arraylist/array or any other interfaces in your program. There's some suggestions on how you can implement the MyDate class. But, you are free to implement it in any way you preferred.

1. In first part of your HW5, you will work on Java Strings. A Java String Array has been provided. This is a chapter from the classic novel "Frankenstein". You need to write some methods to perform some analysis of this chapter. You also need to write a test class (main method) to test your methods. Write a Java Program that will provide the following outputs:

(a) Total number of characters in the chapter
(b) Average number of characters in a word
(c) Total number of words in the chapter
(d) list of all first letter alliterations, i. e. a list of each pair of adjacent words that start with the same letter; this list should not include alliterations on the simple words A, AN, AND, THE, TO, OR, ARE, AM, I, OF, IS, AT, DO, DOES
(e) The character distribution, i. e. a count of the number times each letter is used (example: number of As in the chapter, number of Bs in the chapter, etc.)

Sample Outputs:

Character Count: 6###
Average Word Length: 4.#
Word Count: 1###

First Letter Alliterations:
SHALL SATIATE
HIS HOLIDAY
CANNOT CONTEST
MANY MONTHS
WITH WHICH ...

Character Distribution:
 As: 5##
Bs: 8#
Cs: 1##
Ds: 2##
Es: 8## ...

2. Design a Class named **Person** and its two subclasses named **Student** and **Employee**. Make **Faculty** and **Staff** subclasses of **Employee**. A person has a name, address, phone number, and email address. A student has a class status (freshman, sophomore, junior, or senior). Define the status as a constant. An employee has an office, salary, and date hired. Use the **MyDate** class defined below to create an object for date hired. A faculty member has office hours and a rank. A staff member has a title. Override the **toString** method in each class to display the class name and the person's name.

Write a test program (main method) that creates a Person, Student, Employee, Faculty, and Stuff, and invokes their toString methods.

This is an open-ended assignment. It's upto you what you want to do with these data/how you might want to use these data/how you want to present these data. You might need to guide/tell the user how they will proceed with your program.

Also, you might want to design a user interface. It will present the user a menu. It will ask the user about whose information it needs. Then according to the user's choice, it will provide the information to the user. I am providing a sample o/p below. But, you are not restricted to follow this sample. You can do what you prefer.

Sample O/P:
1-Student
2-Faculty
3-Staff
0-exit

Choose your option:

1

You are about to receive information about student.

Which information do you need?
1-Name
2-Class Status

2

Class Status: Sophomore

The **myDate** Class:

a) The class contains the data fields, year, month and day that represent a date.
b) You can use a method named setDate(long elapsedTime) that sets a new date for the object using the elapsed time.
c) You can use GregorianCalendar class from Java API to display the year, month, and day.

For Section 202: Extension of Problem 1: 20 Points OR Extra Credit for Section 502

Write a method that will provide
a list of all the words used in the chapter (just the words, not the duplications). Also, write the method in that way
so that the unique words are placed in a separate string array. The method then returns this new array, then outputs the list from that array. The method can be void or non-void.