Homework 3

Prerequisites before starting on Homework 3:

1) For those that have done it incorrectly, take out the main method from your **Student** class and put it in a new class. (That means, create a new class and put the main method in it) Name it appropriately.

2) Secondly, if you have done it incorrectly, put all the code you have used to generate data outside the **Student** class. That means, there should NOT be any code inside your **Student** class that randomize the GPA, the venus login, etc.

Instructions for Homework 3:

1) Create a new project called Homework3 and import the 'correct version' (that means you need to make the changes listed in prerequisites above) of Homework 2 into your new project. There should be two new classes and/or two files. One class should be called Student and the other should be called appropriately. I recommend names like StudentGenerator or StudentDriver.

2) Create a new class called **Date**. This class should have three attributes of integer type: **year**, **month**, **day**. In this new class:

i) Create an empty/default constructor (No parameters/arguments). In the constructor, initialize the year, month, and day to **January 1st, 1950**.

ii) Create an **overloaded constructor** that accepts the following parameters in the exact order: (You are not allowed to change the name. Points will be deducted if you don't use this name)

a) **day** (integer type)

b) **month** (integer type)

c) **year** (integer type)

Initialize all of the attributes of the **Date** object using the passed in parameters specified above.

3) Create/Insert a new attribute in your **Student** class called **birthDate** (of type **Date**).

4) Change the **CUNYID** attribute in your Student class to **ID** instead.

5) Lastly, make your **Student** class **abstract**.

6) Create a new class called **QueensCollegeStudent** that 'inherits' from the **Student** class.

[Checkpoint:] By now, you should have a total of 4 classes. (**Date**, **Student**, **QueensCollegeStudent**, **StudentGenerator/Driver** - The class that host your main method)

7) Move the **venusLogin** attribute from the **Student** class to **QueensCollegeStudent** class. (That means, delete **venusLogin** in **Student** class and insert it into the **QueensCollegeStudent** class)

8) Create the get accessors for all your attributes in both classes. E.g., the **getFirstName()** method should return only the first name of the student.

9) Create a new class called **MyInvalidDateException** that inherits from the **Exception** class.

i) Create an empty/default constructor and make it call the base class' constructor

ii) Create an overloaded constructor that accepts a **String** object and make it call the base class' constructor with the same argument type.

[Checkpoint:] At this point you should have a total of 5 classes. (**Date, Student, QueensCollegeStudent, MyInvalidDateException, StudentGenerator/Driver**)

10) Modify the **overloaded constructor** in **Date** class so that it will now check whether the passed in parameters are correct (Recall that each month have different day. The month of January have 31 days while the month of February has 28 days if it is not a leap year and 29 days if it is a leap year. Learn how to formulate this problem appropriately).

Throw a **MyInvalidDateException** if the date is incorrect.

11) In your **Student** class, modify the overloaded constructor so that it accepts the following parameters:

i. firstName, (String type)

ii. lastName, (String type)

iii. ID, (integer type)

iv. GPA, (double type)

v. year, (integer type)

vi. month, (integer type)

vii. day (integer type)

If you have done Homework 2, the only code you need to add is to instantiate a date object for your **birthDate** attribute using year, month and day parameters. Make sure you use the try catch block to handle the **MyInvalidDateException thrown by the Date's overloaded constructor.**

If an invalid date is given and its exception caught, just instantiate a date object using the default/empty constructor.

12) Now, in your **QueensCollegeStudent** class, create only one constructor that accepts the following parameters:

viii. firstName, (String type)

ix. lastName, (String type)

x. ID, (integer type)

xi. GPA, (double type)

xii. year, (integer type)

xiii. month, (integer type)

xiv. day, (integer type)

xv. venusLogin (String type)

In this constructor, call the base class' constructor and pass in all the values EXCEPT **venusLogin**.

Initialize **venusLogin** attribute of the **QueensCollegeStudent** class using the given **venusLogin parameter value.**

[As you are doing this exercise, note that the bulk of the work is in the Student class and the QueensCollegeStudent class is just using what Student has. Think about how tedious and error-prone it is if you have to create a separate student class for different colleges. Also note that venusLogin does not make sense in the abstract Student class but does make sense in the QueensCollegeStudent class]

13) Finally, modify the code in your **main** method to now include the randomization for **birthDate** (**year**, **month** and **day**). At minimum, you should allow one QueensCollegeStudent object to have a birthdate on Feb 29 in a leap year.

Everything else should remain in place (Recall that you were supposed to be doing your randomization of data and construction of **venusLogin** OUTSIDE your **Studen**t class.) Remember to modify your **Student** array object in your main method to now use **QueensCollegeStudent** or else your code will not compile! (Remember, you get a BIG ZERO if it does not compile.)

[Important Note:]

Remember all the object-oriented design principles you are taught in class. Throughout your program, you need to demonstrate that you understand these concepts. That means, you need to decide when or where something should be private, public, or protected. Your logic and design has to be sound. (Recall how it does not make sense that a stranger in the street have access to your 'private' attributes just to get your name. E.g., instead of allowing a stranger to reach out in your pocket/handbag to get your wallet in order to look at your ID, you will instead just give them your name in a different way through a 'public' interface called **SpeakName()**).

Also, we will **reuse the code that you do in Homework 3 for Homework 4**. That means, if you do this right (proper object oriented design), you will have very little changes needed. That means, it pays to design it correctly.

[Professor Note:]

I know the book uses lower camelcase for all methods, but for the purpose of this class, **use Upper CamelCase for all your methods that does not return its actual attributes** so that it will look different whenever you see something that is encapsulated. This should raise an alert to warn you that what you see is not necessarily what it seems underneath. This will greatly help you as you progress into harder problems and during debugging. Without a good system like this, you will spend a lot of time just to locate your problems.