Java For Industry: Lab assignment 2

Question 1 [7 marks]

Write a complete java program for raising and storing invoices. Follow these steps carefully.

1. Create a new class in the programme called Invoice
2. Add five properties to Invoice:
    - *id* of type integer
    - *description* of type String
    - *quantity* of type integer
    - *unitPrice* of type double
    - *paid* of type boolean (true if the invoice Item is paid and false otherwise)

3. Add a four parameter **constructor** to initialise the first four properties of Invoice plus setting paid to false.
4. Add a **toString() method** to return the string representation of the object Invoice as "InvoiceItem[id=?, description=?, quantity=?, uniPrice= ?, paid=?]"
5. create getter and setter methods for paid (getPaid and setPaid).
6. In the Q1Main class, create a main method which declares 2 new instances of Invoice, prints out there contents using the toString method and sets them both to paid.

Question 2 [8 marks]

In this part you will not need to code anything but upload a written answer in your zip file saved as a pdf:

You have been asked to develop a membership system for a local gym and health club. The system allows for members to book into different fitness classes such as spinning or yoga. Membership can be either Junior or Adult. Junior members can't book into adult classes and adults can't book junior classes. Membership can either be gold, silver or bronze tier.
State the name and purpose of 4 classes that might be part of this system, include key methods or variables that are intrinsic to the functioning of the membership system.

Question 3 [10 marks]

In this part you will build a simple password generator and validation class. Alongside some test code. Obviously, this is just a toy example and not something to be used in the real world!

i. Create a program with the classes Password and Q3Main.

ii. In Password write a static method ‘generator’ that takes three arguments; length, symbols and digits. The method should return a secure jumbled up password of the given length with the correct number of symbols and digits. Remaining characters can be given by letters, randomised to be either upper or lower case. **[4 marks]** For example:

length 16, symbols 4 and digits 3 could produce the password ‘gD3Wd$^Skw2d%E\8’

iii. In Password write a second static method to validate the quality of a password passed as a string parameter. The method should return one of the strings in the following table depending on whether the conditions have been met. All conditions must be met for the password to be in the category. **[4 marks]**

	length	Symbols	digits	mix of letter cases
Poor	<= 8	<=1	<=2	FALSE
ok	> 8	>1	>2	FALSE
Good	>12	>3	>3	TRUE
Excellent	>=16	>4	>4	TRUE

iv. In your Q3Main class create a main method. Test your Password class by creating the following 5 passwords and outputting the result of the validation **[2 marks]**:

Characters	Symbols	Digits	Expected Validation
6	2	1	Poor
14	4	5	Good
9	2	3	ok
9	2	2	Poor
18	5	6	Excellent