Java For Industry Mid-Term Assignment

This assignment is out of 60 marks. It is broken down as follows:

Question 1: 25 marks
Question 2: 25 marks
Style and accuracy: 10 marks.

You will receive style and accuracy marks for correctly choosing coding techniques, implementing them well and for organised and commented code.

Please write your answers in the provided IntelliJ template.

The assignment is due on the

Question 1 [25 marks]

In this part you are going to build a small dummy records system for storing student information in a university. Please follow these instructions carefully.

i. Your programme should consist of 3 classes **StudentMain** (This is in the example template), and **Student** and **Grade** that you will need to create yourself**.**

ii. Add the following private properties to Student, you will need to determine the correct type for each variable **[3 marks]**

  - name
  - department
  - age
  - userName
  - studentNumber
  - fullTime (is the student full or part-time)

iii. Create getters and setters for each of the private instance variables. **[2 marks]**

iv. Write a constructor that takes arguments for name department, age, student number and fullTime.
[2 marks]

v. In your constructor, assign a user name to the student. It should be constructed with their first name initial, the first 4 letters of their surname followed by the first 3 digits of their student number. Therefore a student called Bert Smith, student number 12345 would have the user name ‘bsmit123’. Store the user name in the userName variable **[5 marks]**

vi. Add a public ArrayList to the student class called grades. grades should store objects of the type Grade. **[1 mark]**

vii. Add 2 private instance variables to Grade; subject and score. Create a constructor that sets both these values, and getter and setter methods for each. **[2 marks]**

viii. Add a static method to Grade, called getLetterGrade, that returns a lettered grade of type char for a parameter score of type double. So that **[3 marks]**:

- a score of >= 70% is an A
- a score of >= 60% is a B
- a score of >= 50% is a C
- a score of >= 40% is a D
- a score of <40% is an F
- An invalid score (either less than 0 or greater than 100 returns 'E" for error.

ix. In the main method of your Main class create the following students and grades using the code written above in an ArrayList. The arrayList should be declared as a private static variable of the Main class (You can just add grades one by one to each students grade list directly) **[2 marks]**

NAME	DEPARTMENT	AGE	USER NAME	STUDENT NUMBER	FULLTIME?	GRADES
Bert Smith	computing	21	bsmit123	12345	TRUE	programming 52, web dev 63, maths 76, algorithms 68
Olivia Green	computing	19	ogree234	23464	TRUE	programming 73, web dev 82, maths 72, algorithms 66
Eloise Jones	computing	18	ejone347	34744	TRUE	programming 65, web dev 63, maths 37, algorithms 40
Ben Bird	computing	42	bbird348	34834	FALSE	programming 55, web dev 29, maths 56, algorithms 38
Karen Brown	computing	25	kbrow456	45632	FALSE	programming 62, web dev 51, maths 43, algorithms 43

x. Create a menu system in the Main class that allows for uses to interrogate the system. It should have the following options **[5 marks]**.

- Print out a report of all students including lettered grades for each module score
- Print the name of all students with a failed module
- print out average grade for each student
- Add a new student to the system taking in name, department, age, student number and a grade for each of the four modules.

- Quit the program.

Question 2 [25 marks]

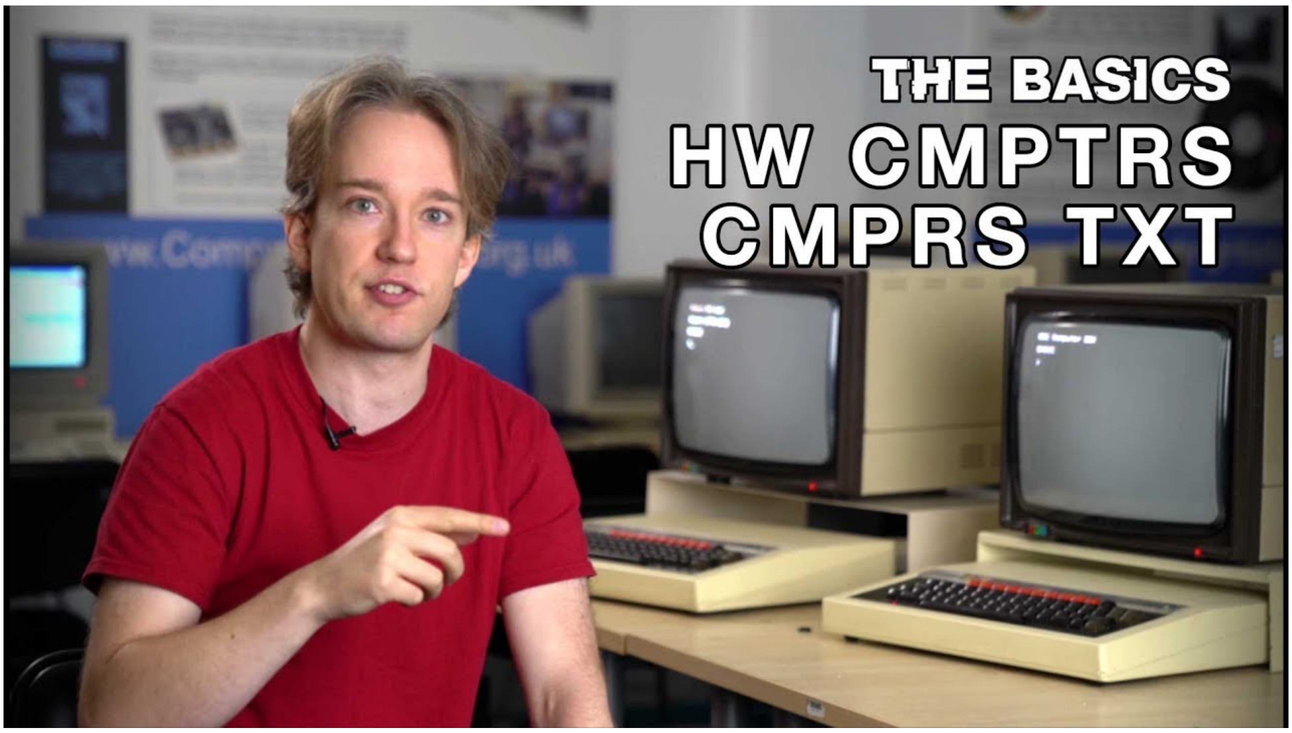
In this question you will read in a file from the computers disk drive. The file, metamorphosis.txt is provided for you. It is important that you do not change the contents of the file so we can accurately mark your work.

i. Read in the file that has provided for you in the template using the Scanner class. **[3 marks]**
ii. Output the following statistics from the file to the terminal. **[5 marks]**
   - Length in characters
   - Number of words
   - Number of sentences
   - Number of paragraphs
iii. Create a word frequency list for the text (how many times each word appears) save your list into a new file called frequencies.txt **[7 marks]**. To complete this successfully, you will need to determine what counts as a different word. Apply the following rules:
   - cat and cats are different words but
   - cat and cat's are the same word
   - cat. and cat are the same, as is
   - cat! cat? and cat
   - can't and cannot are different words
iv. Extension: Generate Huffman codes for the text **[10 marks]**

Only attempt this final part when you have done the others. It is designed to be challenging!

You answer must utilise two new classes, Tree, to construct and store the Huffman codes and Node, to represent each path or letter through the tree. In your answer represent the final code as a string of ones and zeros.

Here is a link to a video that describes Huffman trees if you haven't come across them before:


THE BASICS
HW CMPTRS
CMPRS TXT