Classes and Objects

Case Study

For the current academic year, a batch of 65 students has been enrolled for Native programming module. The total module mark is derived from marks obtained for 05 components specified below.

- Minor Assignment (15%)
    - Which is made up of two sub components
        - Minor Assignment 01 – 6%
        - Minor Assignment 02 – 9%
- Project (35%)
    - Which is made up of two sub components
        - Design – 15%
        - Implementation – 20%
- Final Exam (50%)

Each of the above assignment components is marked out of 100 and their contribution percentages for the total module mark are as follows.

- Minor Assignment – 15%
- Project – 35%
- Final Exam – 50%

The students' final grades are decided as follows.

Range	Grading
100 to 80	Higher Distinction
79 to 70	Distinction
69 to 60	Credit
59 to 50	Pass
Below 50	Fail

The grading procedure also considers the following facts.

a) To pass the module it is mandatory student obtains at least 50% in the final exam

b) Also, the student must get minimum of 35% in the Minor Assignment and Project component

c) Also, the student must get minimum of 50% as overall marks for the module

d) Only those who successfully meet the criteria (a), (b) and (c) will be awarded an appropriate grading.

e) A student will have to obtain a total module mark of 50% in order to avoid re-taking the module. (Failing to obtain at least 50 will result in a re-take.)

f) If the student has obtained a total module mark above 50%, but has failed to achieve a minimum of 50% in the final exam will be given a supplement exam and their grades will be recorded as **Incomplete** – "I"

You are required to,

Write a Java program (**Command Line Interface Application**) with functional decomposition, using necessary validations for inputs and appropriate good coding practices for the following expectations of the customer:

1. Accept the students' details (Application should be capable of any given number) and store them in an appropriate form (choose the most suitable data structure to store the input captured through keyboard). **Hint:-** see the usage of ArrayList.

2. Create appropriate user-defined data types and the details of the students should include but need not to be limited to:
    - First Name
    - Last Name
    - Registration No.
    - Marks obtained for each component/subcomponent

3. Calculate the component marks and store them in a suitable instance variable where there are subcomponents (e.g. Major Assignment and project)
4. Calculate the overall marks and store them in suitable instance variables
5. Determine the grades and store them in a suitable instance variable

---

6. Find the average for each component separately and display them based on the marks obtained by all the students enrolled.

7. Find the highest scorer for each component and overall module separately and display those students' details with suitable headings.

8. Find the list students who are eligible for supplement exam

9. Display the list of students who have obtained retake for the module

10. Display the list of students in the descending order of their overall marks along with categorization under their grades for only those who passed