Object-Oriented Programming with Java: Practical Assignment 1

Marks: The assignment carries 10% marks and it will be added to the final examination.

Instructions

1. Read the instructions carefully to understand the concepts tested in the question. Refer notes/slides for additional information.
2. Use NetBeans IDE and execute each of the programs.
3. You must submit handwritten answers as an assignment submission (No hard copy is accepted). Hence, after successfully executing each of the programs, write the answer in a sheet.
4. Follow the same with all the questions.
5. Write your index number, name and batch name clearly on the front page and handover the hand-written assignment on the 13th of November 2019(Wednesday) noon, before the commencement of the lecture.

Questions

1. Open a new project using Netbeans IDE and provide the Project Name as 'StudentObj'. Within the main method write a program to display the following output.

*

**

***

****

*****

2. Remove the contents of the main method. In the same StudentObj project add a new class 'Student'.

Include the following 'Data' and 'Methods' inside the same 'Student' class.

Data : Student No (integer) , Student Name (String) , Both of them are private variables

Methods : 1. A method to set Values to the above two variables

2. A method to display values of the above variables

Both above methods are public

3. Main method is still available in the StudentObj class. In the StudentObj class within the main method write code to Create an object from the Student class and call the above two methods.

4. Create another object from the same class and call the methods.

5. Remove Method 1 & 2 in the Student class. Include two ‘setter’ function to set values to ‘Student No’ and ‘Student Name’. Include two ‘getter’ function to return the values.

6. In the ‘StudentObj’ class and remove the content of main method. Now create object from the ‘Student’ class and call the ‘setter’ and ‘getter’ functions.

7. Remove the ‘getter’ and ‘setter’ functions from the ‘Student’ class. Include ‘no parameter’ and ‘parameterized constructor’ into the ‘Student’ class.

8. Amend the main method in ‘StudentObj’ class to create an object from the ‘Student’ class and call no parameter and parameterized constructors.

9. Change the access modifiers of ‘Student’ class ‘Student No’ and ‘Student Name’ into ‘protected’.

10. In the ‘StudentObj’ project add a new class ‘Marks’. The ‘Marks’ class contains ‘module1’ and ‘module2’ as data (integer) and display method to display ‘Student No’, ‘Student Name’ ,‘marks1’and ’marks2’.

11. Include ‘no parameter constructor’ and ‘parameterized constructor’ into ‘Marks’ class.

12. Create an object from the ‘Marks’ class and call the parameterized constructor of ‘Marks’ class and it’s display method.

13. In the ‘StudentObj’ project include an abstract class ‘Grade’. The abstract class ‘Grade’ contains an abstract method ‘displayGrade’ with void as return type.

14. The abstract class ‘Grade’ method ‘displayGrade’ should be implemented inside the ‘Marks’ class. The logic to implement the ‘displayGrade’ method as follows.

Calculate the average of ‘Module 1’ and ‘Module 2’. If the average is above 50 display as ‘pass’ and otherwise as ‘fail’.

15. Inside the ‘StudentObj’ edit the main method and display the grade.

16. Inside the ‘StudetObj’ include an interface named ‘BonusMarks’ with constant integer variable m with the value 10. Include a method ‘provideMarks’ inside the ‘BonusMarks’ interface.

17. Write the ‘Marks’ class to extend from the already created abstract class ‘Grade’ and implement the ‘BonusMarks’ interface. Implement the ‘provideMarks’ method to display the following message.

“10 marks will be provided to the continuous presence to the lectures”

18. Amend the ‘StudentObj’ main method to create an object from the ‘Marks’ class and call all the possible methods.