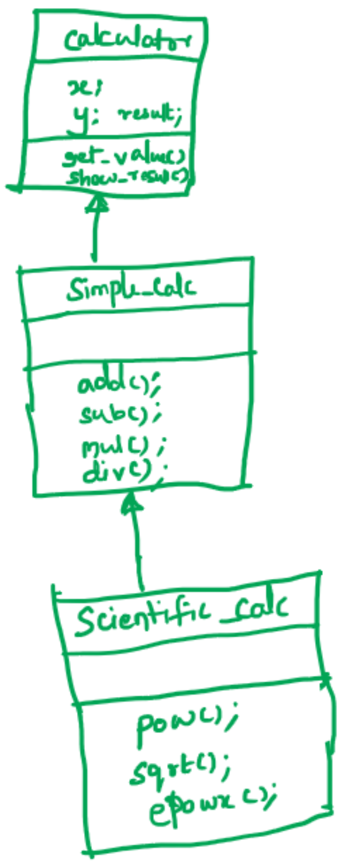
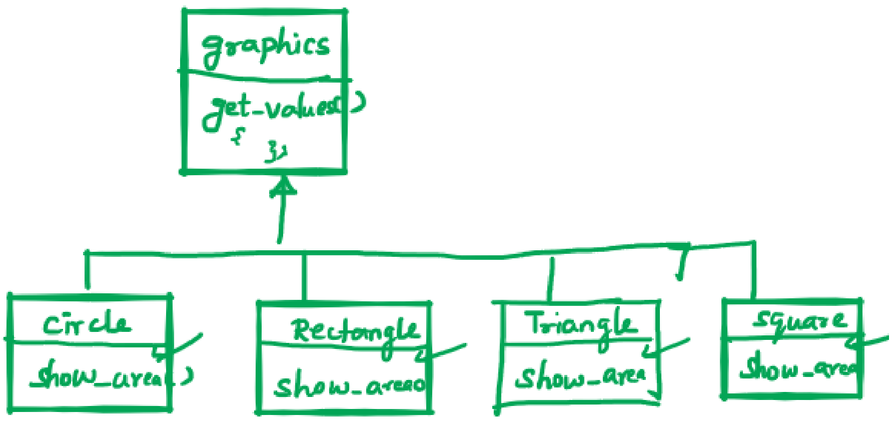
1. Create a JAVA program for the below inheritances (Multilevel and Hierarchical):





2. Create a base class Fruit with name, taste and size as its attributes. Create a method called eat() which describes the name of the fruit and its taste. Inherit the same in 2 other classes Apple and Orange and override the eat() method to represent each fruit taste.
3. Create a class named 'Animal' which includes methods like eat() and sleep(). Create a child class of Animal named 'Bird' and override the parent class methods. Add a new method named fly(). Create an instance of Animal class and invoke the eat and sleep methods using this ob ject. Create an instance of Bird class and invoke the eat, sleep and fly methods using this object.

4. Create a superclass, Student, and two subclasses, Undergrad and Grad. The superclass Student should have the following data members: name, ID, grade, age, and address.The superclass, Student should have one method: boolean isPassed (double grade) The purpose of the isPassed method is to take one parameter, grade (value between 0 and 100) and check whether the grade has passed the requirement for passing a course.In the Student class this method should be empty as an abstract method. The two subclasses: Grad and Undergrad, will inherit all data members of the Student class and override the method isPassed. For the UnderGrad class, if the grade is above 70.0, then isPassed returns true, otherwise it returns false. For the Grad class, if the grade is above 80.0, then isPassed returns true, otherwise returns false. Create a test class for your three classes. In the test class, create one Grad object and one Undergrad object. For each object, provide a grade and display the results of the isPassed method.
5. Write a Java program which has Interface class for Stack Operations.
   (i) A Class that implements the Stack Interface and creates a fixed length Stack.
   (ii) A Class that implements the Stack Interface and creates a Dynamic length Stack.
6. Write a Java program using Synchronized Threads, which demonstrates Producer Consumer concepts.