**1. Packages & Import Statements**

* **Task:**
  1. Create a package named com.assignment.shapes.
  2. Inside this package, create a class Rectangle with instance variables length and breadth (private) and a method area() (public).
  3. Create another package com.assignment.main, where you import the Rectangle class and use it in the Main class.
  4. Compile and run the program.

**2. Access Modifiers Practice**

* **Task:**
  1. Create a class Person with the following instance variables:
     + name (public)
     + age (private)
     + address (default)
  2. Create appropriate getter and setter methods for age.
  3. Create a Main class to access and modify the variables, ensuring proper encapsulation.

**3. Inheritance Implementation**

* **Task:**
  1. Create a base class Animal with a method makeSound() that prints "Some generic sound".
  2. Create a subclass Dog that overrides makeSound() to print "Bark!".
  3. Create another subclass Cat that overrides makeSound() to print "Meow!".
  4. In the Main class, create objects of Dog and Cat and call makeSound().

**4. Polymorphism - Method Overloading**

* **Task:**
  1. Create a class Calculator with overloaded methods add():
     + add(int a, int b)
     + add(double a, double b, double c)
     + add(String a, String b)
  2. In the Main class, call all versions of add() and observe the results.

**5. Polymorphism - Method Overriding**

* **Task:**
  1. Create a base class Vehicle with a method fuelType() that returns "Generic Fuel".
  2. Create subclasses Car and Bike, overriding fuelType() to return "Petrol" and "Diesel" respectively.
  3. Create a Main class where you use method overriding by creating objects of Car and Bike and calling fuelType().

**Bonus Task (Advanced)**

* **Task:**
  1. Create a BankAccount class with:
     + A private variable balance
     + A public method deposit(double amount)
     + A protected method withdraw(double amount)
  2. Create a subclass SavingsAccount that overrides withdraw() to ensure withdrawal is only allowed if the balance is sufficient.
  3. Test the functionality in the Main class.