1. Create a base class BankAccount with methods like deposit() and withdraw(). Derive a class SavingsAccount that overrides the withdraw() method to impose a limit on the withdrawal amount. Write a program that demonstrates the use of overridden methods and proper access modifiers & return the details.

```
package bank;

public class BankAccount {
protected double balance;

public BankAccount(double initialBalance) {
this.balance = initialBalance;
}

public void deposit(double amount) {
balance += amount;
}

public void withdraw(double amount) {
if (amount <= balance) {
balance -= amount;
} else {
System.out.println("Insufficient balance.");
}
}

public double getBalance() {
return balance;
   }
}



package bank;

public class SavingsAccount extends BankAccount {
private double withdrawalLimit;

public SavingsAccount(double initialBalance, double limit) {
```

```java
super(initialBalance);
this.withdrawalLimit = limit;
}

@Override
public void withdraw(double amount) {
if (amount <= withdrawalLimit && amount <= balance) {
balance -= amount;
} else {
System.out.println("Withdrawal amount exceeds limit or insufficient
balance.");
    }
  }
}
```

```java
package bank;

public class TestBankAccount {

public static void main(String[] args) {
SavingsAccount account = new SavingsAccount(5000, 1000);
account.deposit(1500);
System.out.println("Balance: " + account.getBalance());
account.withdraw(800);
System.out.println("Balance after withdrawal: " +
account.getBalance());
account.withdraw(1200);
}
}
```

OUTPUT:

```
Balance: 6500.0
Balance after withdrawal: 5700.0
Withdrawal amount exceeds limit or insufficient balance.
```

2. Create a base class Vehicle with attributes like make and year. Provide a constructor in Vehicle to initialize these attributes. Derive a class Car that has an additional attribute model and write a constructor that initializes make, year, and model. Write a program to create a Car object and display its details.

```
package com.vehicles;

public class Vehicle {
protected String make;
protected int year;

public Vehicle(String make, int year) {
this.make = make;
this.year = year;
}

public void displayDetails() {
System.out.println("Make: " + make + ", Year: " + year);
   }
}


package com.vehicles;

public class Car extends Vehicle {
private String model;

public Car(String make, int year, String model) {
super(make, year);
this.model = model;
}

@Override
public void displayDetails() {
System.out.println("Make: " + make + ", Year: " + year + ", Model: " +
model);
```

```
   }
}
```

```
package com.vehicles;

public class TestVehicle {

public static void main(String[] args) {
Car car = new Car("Toyota", 2024, "Corolla");

car.displayDetails();
}
}
```

OUTPUT:

```
Make: Toyota, Year: 2024, Model: Corolla
```

3. Create a base class Animal with attributes like name, and methods like eat() and sleep(). Create a subclass Dog that inherits from Animal and has an additional method bark(). Write a program to demonstrate the use of inheritance by creating objects of Animal and Dog and calling their methods.

```
package com.animals;

public class Animal {
protected String name;

public Animal(String name) {
this.name = name;
}

public void eat() {
System.out.println(name + " is eating.");
}

public void sleep() {
System.out.println(name + " is sleeping.");
  }
}
```

```
package com.animals;

public class Dog extends Animal {
 public Dog(String name) {
 super(name);
 }

 public void bark() {
 System.out.println(name + " is barking.");
  }
}
```

```
package com.animals;

public class TestAnimal {

public static void main(String[] args) {
 Animal genericAnimal = new Animal("Animal");

 Dog dog = new Dog("Pebbles");

 genericAnimal.eat();
 genericAnimal.sleep();

 dog.eat();
 dog.sleep();
 dog.bark();
}
}
```

OUTPUT:

```
Animal is eating.
Animal is sleeping.
Pebbles is eating.
Pebbles is sleeping.
Pebbles is barking.
```

4. Build a class Student which contains details about the Student and compile and run its instance.

```
package com.studentdetails;

public class Student {
private String name;
private int age;
private String rollNumber;
private String course;

public Student(String name, int age, String rollNumber, String course)
{
this.name = name;
this.age = age;
this.rollNumber = rollNumber;
this.course = course;
}

public void displayDetails() {
System.out.println("Student Name: " + name);
System.out.println("Age: " + age);
System.out.println("Roll Number: " + rollNumber);
System.out.println("Course: " + course);
}

public static void main(String[] args) {
```

```
Student student = new Student("Janvi Kumari", 18, "4001", "Computer
Science");
student.displayDetails();
}
}
```

OUTPUT:

```
Student Name: Janvi Kumari
Age: 18
Roll Number: 4001
Course: Computer Science
```

5. Write a Java program to create a base class Vehicle with methods startEngine() and stopEngine(). Create two subclasses Car and Motorcycle. Override the startEngine() and stopEngine() methods in each subclass to start and stop the engines differently.

```
package com.vehicles;

public class Vehicle {
public void startEngine() {
System.out.println("Starting the vehicle engine...");
}

public void stopEngine() {
System.out.println("Stopping the vehicle engine...");
}
}
```

```
package com.vehicles;

public class Car extends Vehicle {

@Override
```

```java
public void startEngine() {
System.out.println("Starting the car engine with a key.");
}

@Override
public void stopEngine() {
System.out.println("Stopping the car engine by turning off the key.");
}
}
```

```java
package com.vehicles;

public class Motorcycle extends Vehicle {

@Override
public void startEngine() {
System.out.println("Starting the motorcycle engine with a kick
start.");
}

@Override
public void stopEngine() {
System.out.println("Stopping the motorcycle engine by pressing the
kill switch.");
}
}
```

```java
package com.vehicles;

public class TestVehicle {

public static void main(String[] args) {
Car car = new Car();
car.startEngine();
car.stopEngine();

System.out.println("---------------------------");
Motorcycle motorcycle = new Motorcycle();
motorcycle.startEngine();
motorcycle.stopEngine();
}
}
```

OUTPUT:

```
Starting the car engine with a key.
Stopping the car engine by turning off the key.
---------------------------
Starting the motorcycle engine with a kick start.
Stopping the motorcycle engine by pressing the kill switch.
```