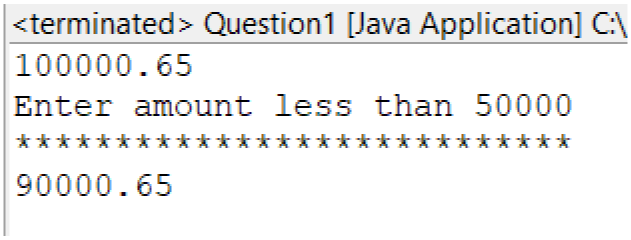
OOPJ Assignment 5

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 com.assignment.question1;

class BankAccount{
    private double amount;
    public void deposit(double amount) {
        this.amount += amount;
    };

    public void withdraw(double amount){
        this.amount -= amount;
    }

    public double getAmount() {
        return amount;
    }
}

class SavingsAccount extends BankAccount{
    public void withdraw(double amount) {
        if(amount<50000) {
            super.withdraw(amount);
        }
        else {
            System.out.println("Enter amount less than 5000
0");
        }
    }
}

public class question1 {

    public static void main(String[] args) {
        BankAccount savings = new SavingsAccount();  //Upcast
ing
        savings.deposit(100000.65);
        System.out.println(savings.getAmount());
        savings.withdraw(60000);

        System.out.println("****************************");

        savings.withdraw(10000);
        System.out.println(savings.getAmount());

    }
}
```

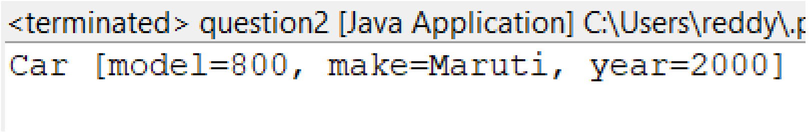


```
<terminated> Question1 [Java Application] C:\
100000.65
Enter amount less than 50000
****************************
90000.65
```

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.

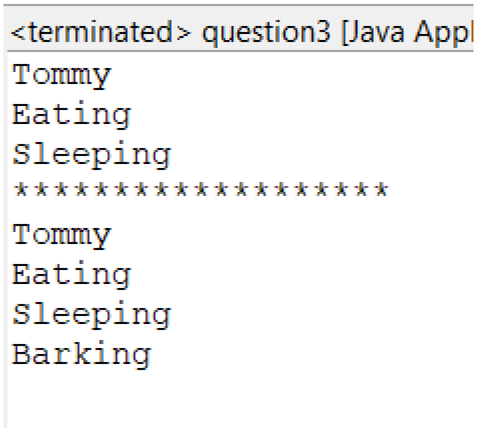
```
package com.assignment.question2;

class Vehicle{
    String make;
    int year;

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

}

class Car extends Vehicle{
    String model;

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

    @Override
    public String toString() {
        return "Car [model=" + model + ", make=" + make +
", year=" + year + "]";
    }

}

public class question2 {

    public static void main(String[] args) {
        Car car = new Car("Maruti", 2000, "800");

        System.out.println(car.toString());

    }
}
```



```
<terminated> question2 [Java Application] C:\Users\reddy\.p
Car [model=800, make=Maruti, year=2000]
```

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.assignment.question3;

class Animal{
    String name="Tommy";

    public void eat() {
        System.out.println("Eating");
    }

    public void sleep() {
        System.out.println("Sleeping");
    }
}

class Dog extends Animal{
    public void bark() {
        System.out.println("Barking");
    }
}

public class question3 {
    public static void main(String[] args) {
    Animal animal = new Animal();

    System.out.println(animal.name);
    animal.eat();
    animal.sleep();
    //animal.bark(): will produce a error as there is no b
ark method in animal
    System.out.println("*******************");
    Dog dog = new Dog();

    System.out.println(dog.name); //Inherited
    dog.eat();  //Inherited
    dog.sleep();  //Inherited
    dog.bark();  //Inherited
    }
}
```

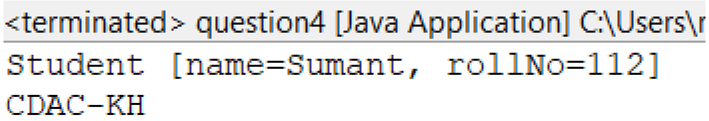


```
<terminated> question3 [Java Appl
Tommy
Eating
Sleeping
*******************
Tommy
Eating
Sleeping
Barking
```

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

instance

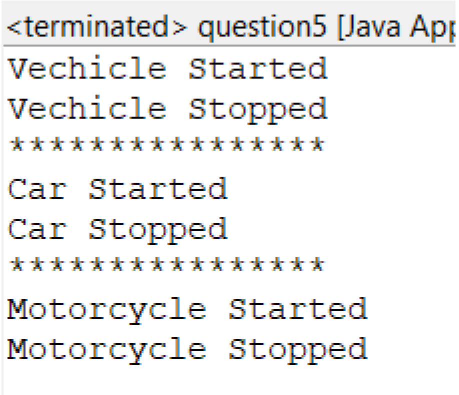
```
package com.assignment.question4;

class Student{
    private String name;
    private int rollNo;
    private static String schoolName = "CDAC-Mumbai";

    Student(){
        this("Not Entered", -1);
    }

    public Student(String name, int rollNo) {
        this.name = name;
        this.rollNo = rollNo;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getRollNo() {
        return rollNo;
    }

    public void setRollNo(int rollNo) {
        this.rollNo = rollNo;
    }

    public static String getSchoolName() {
        return schoolName;
    }

    public static void setSchoolName(String schoolName) {
        Student.schoolName = schoolName;
    }

    @Override
    public String toString() {
        return "Student [name=" + this.name + ", rollNo="
+ this.rollNo + "]";
    }


}

public class question4 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Student student = new Student();

        student.setName("Sumant");
        student.setRollNo(112);
        student.setSchoolName("CDAC-KH");

        System.out.println(student.toString());
        System.out.println(Student.getSchoolName());
    }
}
```



```
<terminated> question4 [Java Application] C:\Users\r
Student [name=Sumant, rollNo=112]
CDAC-KH
```

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.assignment.question5;

class Vehicle{
    public void startEngine() {
        System.out.println("Vechicle Started");
    }

    public void stopEngine() {
        System.out.println("Vechicle Stopped");
    }

}

class Motorcycle extends Vehicle{

    @Override
    public void startEngine() {
        System.out.println("Motorcycle Started");
    }

    @Override
    public void stopEngine() {
        System.out.println("Motorcycle Stopped");
    }
}

class Car extends Vehicle{
    @Override
    public void startEngine() {
        System.out.println("Car Started");
    }
    @Override
    public void stopEngine() {
        System.out.println("Car Stopped");
    }
}

public class question5 {
    public static void main(String[] args) {
        Vehicle vehicle = new Vehicle();
        Car car = new Car();
        Motorcycle moto = new Motorcycle();

        vehicle.startEngine();
        vehicle.stopEngine();

        System.out.println("****************");

        car.startEngine();
        car.stopEngine();

        System.out.println("****************");

        moto.startEngine();
        moto.stopEngine();

    }
}
```



```
Vechicle Started
Vechicle Stopped
****************
Car Started
Car Stopped
****************
Motorcycle Started
Motorcycle Stopped
```