CDAC Mumbai PG-DAC August 24

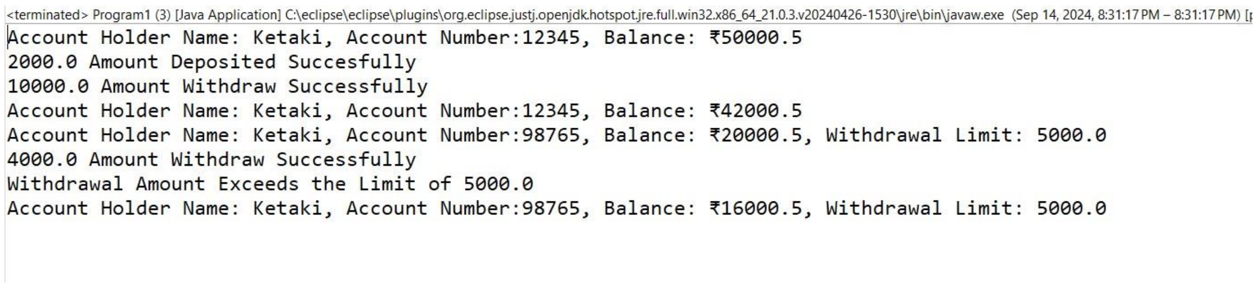
Assignment No- 5

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.

   CODE:

```
package org.example;


class BankAccount{
    private String accountHolderName;
    private int accountNumber;
    private float balance;



    public BankAccount(String accountHolderName, int accountNumber, float initialBalance) {
        this.accountHolderName = accountHolderName;
        this.accountNumber = accountNumber;
        this.balance = initialBalance;
    }

    public void deposit(float amount) {
        if(amount > 0) {
                balance += amount;
                System.out.println(amount+" Amount Deposited Succesfully");
        }else {
                System.out.println("Invalid Amount");
        }
    }

    public void withdraw(float amount) {
        if(amount > 0 && balance >= amount) {
                balance -= amount;
                System.out.println(amount+" Amount Withdraw Successfully");
        }else {
                System.out.println("Insufficient Balance");
        }
    }

    public float getBalance(){
        return balance;
    }

    public String getAccountDetails() {
        return "Account Holder Name: "+ accountHolderName + ", Account Number:" + accountNumber
    + ", Balance: ₹" + balance;
    }


}

class SavingAccount extends BankAccount{

    private float withDrawLimit;

    public SavingAccount(String accountHolderName, int accountNumber, float balance , float
    withDrawLimit) {

        super(accountHolderName , accountNumber , balance);
        this.withDrawLimit = withDrawLimit;
    }

    public void withdraw(float amount) {
        if(amount > withDrawLimit) {
                System.out.println("Withdrawal Amount Exceeds the Limit of "+withDrawLimit);
        }else {
                super.withdraw(amount);
        }
    }

    public String getAccountDetails() {
        return super.getAccountDetails()+", Withdrawal Limit: "+withDrawLimit;
    }

}

public class Program1 {

    public static void main(String[] args) {
        BankAccount b = new BankAccount("Ketaki", 12345 , 50000.50f);
        System.out.println(b.getAccountDetails());
        b.deposit(2000);
        b.withdraw(10000.0f);
        System.out.println(b.getAccountDetails());

        SavingAccount s = new SavingAccount("Ketaki", 98765, 20000.50f, 5000);
        System.out.println(s.getAccountDetails());
        s.withdraw(4000);
        s.withdraw(10000);
        System.out.println(s.getAccountDetails());
    }

}
```

OUTPUT:



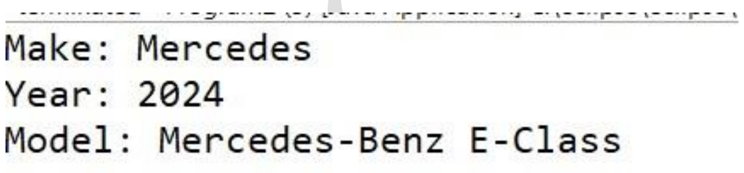
```
<terminated> Program1 (3) [Java Application] C:\eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v20240426-1530\jre\bin\javaw.exe  (Sep 14, 2024, 8:31:17 PM – 8:31:17 PM) [
Account Holder Name: Ketaki, Account Number:12345, Balance: ₹50000.5
2000.0 Amount Deposited Succesfully
10000.0 Amount Withdraw Successfully
Account Holder Name: Ketaki, Account Number:12345, Balance: ₹42000.5
Account Holder Name: Ketaki, Account Number:98765, Balance: ₹20000.5, Withdrawal Limit: 5000.0
4000.0 Amount Withdraw Successfully
Withdrawal Amount Exceeds the Limit of 5000.0
Account Holder Name: Ketaki, Account Number:98765, Balance: ₹16000.5, Withdrawal Limit: 5000.0
```

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.

CODE:

```
package org.example;


class Vehicle1{
    private String make;
    private int year;

    public Vehicle1(String make , int year) {
        this.make = make;
        this.year = year;
    }

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

class Car1 extends Vehicle1{
    private String model;

    public Car1(String make , int year, String model) {
        super(make, year);
        this.model = model;
    }

    public void displayDetails() {
        super.getDetails();
        System.out.println("Model: "+model);
    }
}

public class Program2 {

    public static void main(String[] args) {
        Car1 c = new Car1("Mercedes", 2024, "Mercedes-Benz E-Class");

        c.displayDetails();

    }
}
```

OUTPUT:



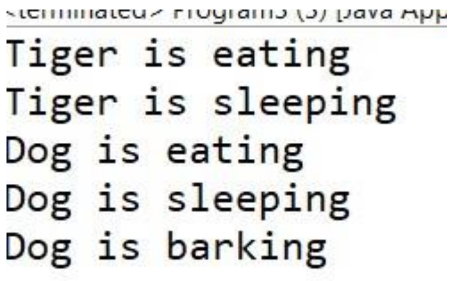
```
Make: Mercedes
Year: 2024
Model: Mercedes-Benz E-Class
```

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.

CODE:

```
package org.example;

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

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

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

class Dog extends Animal{

    public Dog(String name) {
        super(name);
    }

    public void barks() {
        super.eat();
        super.sleep();
        System.out.println(name+" is barking");
    }
}

public class Program3 {

    public static void main(String[] args) {
        Animal animal = new Animal("Tiger");
        animal.eat();
        animal.sleep();
        Dog dog = new Dog("Dog");
        dog.barks();
    }
}
```

OUTPUT:



```
<terminated> Programs (5) [Java App
Tiger is eating
Tiger is sleeping
Dog is eating
Dog is sleeping
Dog is barking
```

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

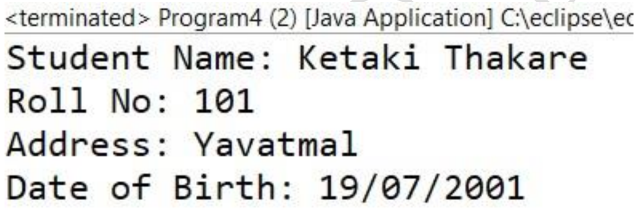
CODE:

```
package org.example;


class Student{
    private String name;
    private int rollno;
    private String address;
    private String dob;

    public Student(String name, int rollno, String address, String dob) {
        this.name = name;
        this.rollno = rollno;
        this.address = address;
        this.dob = dob;
    }
```

```
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 String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}

public String getDob() {
    return dob;
}

public void setDob(String dob) {
    this.dob =dob;
}
```

```
    public void displayDetails() {
        System.out.println("Student Name: "+name);
        System.out.println("Roll No: "+rollno);
        System.out.println("Address: "+address);
        System.out.println("Date of Birth: "+dob);
    }


}

public class Program4 {
    public static void main(String[] args) {
        Student student = new Student("Ketaki Thakare", 101, "Yavatmal", "19/07/2001" );

        student.displayDetails();
    }
}
```

OUTPUT:



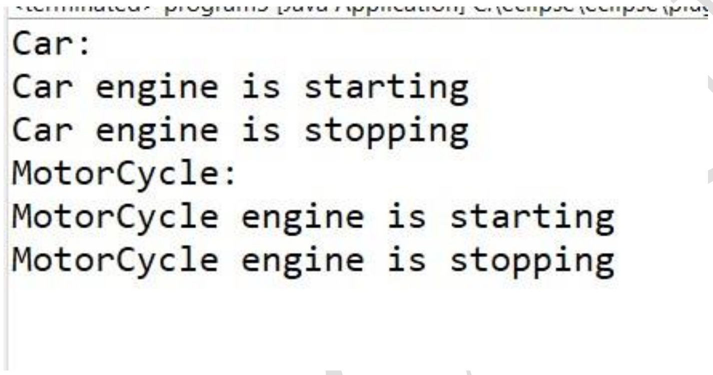
```
<terminated> Program4 (2) [Java Application] C:\eclipse\ec
Student Name: Ketaki Thakare
Roll No: 101
Address: Yavatmal
Date of Birth: 19/07/2001
```

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.

CODE:

```
package org.example;


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

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

class Car extends Vehicle{
    public void startEngine() {
        System.out.println("Car engine is starting");
    }

    public void stopEngine() {
        System.out.println("Car engine is stopping");
    }
}

class MotorCycle extends Vehicle{
    public void startEngine() {
        System.out.println("MotorCycle engine is starting");
    }

    public void stopEngine() {
        System.out.println("MotorCycle engine is stopping");
    }
}
public class Program5 {
    public static void main(String[] args) {
        Vehicle car = new Car();
        System.out.println("Car:");
        car.startEngine();
        car.stopEngine();

        Vehicle motorCycle = new MotorCycle();
        System.out.println("MotorCycle:");
        motorCycle.startEngine();
        motorCycle.stopEngine();
    }
}
```

OUTPUT:



```
Car:
Car engine is starting
Car engine is stopping
MotorCycle:
MotorCycle engine is starting
MotorCycle engine is stopping
```