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.

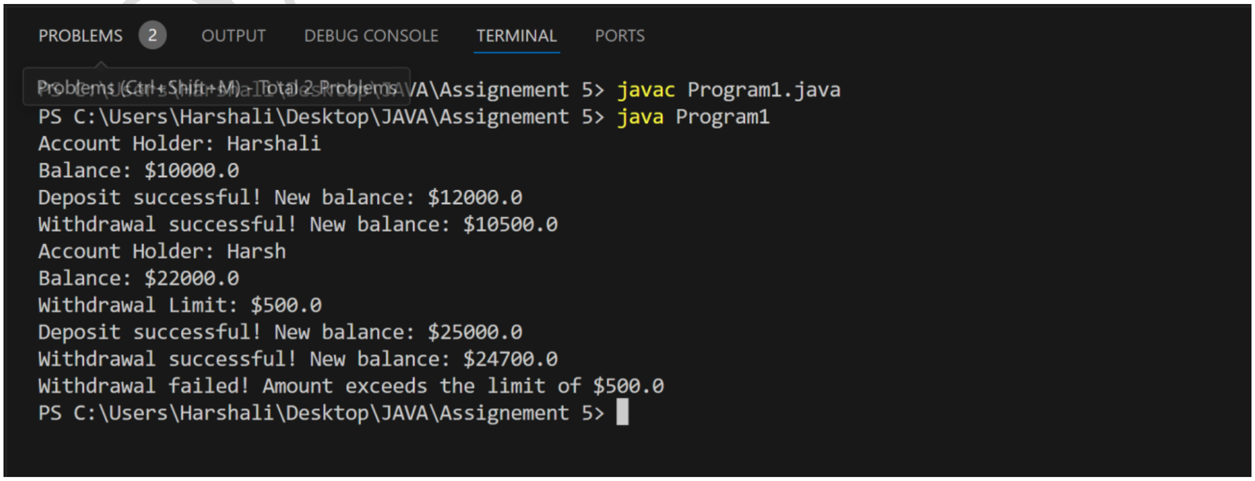
Code:

```
// Base class: BankAccount
class BankAccount {
    protected String accountHolderName;
    protected double balance;

    // Constructor
    public BankAccount(String accountHolderName, double balance) {
        this.accountHolderName = accountHolderName;
        this.balance = balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposit successful! New balance: $" + balance);
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful! New balance: $" + balance);
        } else {
            System.out.println("Insufficient balance or invalid amount.");
        }
    }

    public String getAccountDetails() {
        return "Account Holder: " + accountHolderName + "\nBalance: $" + balance;
    }
}

class SavingsAccount extends BankAccount {
    private double withdrawalLimit;

    // Constructor
    public SavingsAccount(String accountHolderName, double balance, double withdrawalLimit) {
```

CDAC MUMBAI

```
        super(accountHolderName, balance);
        this.withdrawalLimit = withdrawalLimit;
    }
    @Override
    public void withdraw(double amount) {
        if (amount > withdrawalLimit) {
            System.out.println("Withdrawal failed! Amount exceeds the limit of $" + withdrawalLimit);
        } else {
            super.withdraw(amount);
        }
    }

    @Override
    public String getAccountDetails() {
        return super.getAccountDetails() + "\nWithdrawal Limit: $" + withdrawalLimit;
    }
}

public class Program1 {
    public static void main(String[] args) {
        BankAccount bankAccount = new BankAccount("Harshali", 10000);
        System.out.println(bankAccount.getAccountDetails());
        bankAccount.deposit(2000);
        bankAccount.withdraw(1500);

        SavingsAccount savingsAccount = new SavingsAccount("Harsh", 22000, 500);
        System.out.println(savingsAccount.getAccountDetails());


        savingsAccount.deposit(3000);
        savingsAccount.withdraw(300);
        savingsAccount.withdraw(1000);
    }
}
```

DAC MUMBAI



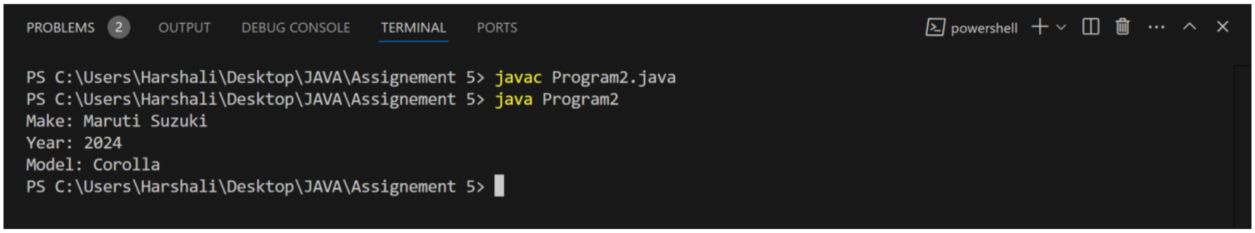
```
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5> javac Program1.java
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5> java Program1
Account Holder: Harshali
Balance: $10000.0
Deposit successful! New balance: $12000.0
Withdrawal successful! New balance: $10500.0
Account Holder: Harsh
Balance: $22000.0
Withdrawal Limit: $500.0
Deposit successful! New balance: $25000.0
Withdrawal successful! New balance: $24700.0
Withdrawal failed! Amount exceeds the limit of $500.0
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5>
```

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:

```
// Base class: Vehicle
class Vehicle {
   protected String make;
   protected int year;

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

   public String getDetails() {
      return "Make: " + make + "\nYear: " + year;
   }
}

// Derived class: Car
class Car extends Vehicle {
   private String model;

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

   @Override
   public String getDetails() {
      return super.getDetails() + "\nModel: " + model;
   }
}

public class Program2 {
   public static void main(String[] args) {
      Car car = new Car("Maruti Suzuki", 2024, "Corolla");
      System.out.println(car.getDetails());
   }
}
```

CDAC MUMBAI



```
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5> javac Program2.java
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5> java Program2
Make: Maruti Suzuki
Year: 2024
Model: Corolla
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5>
```

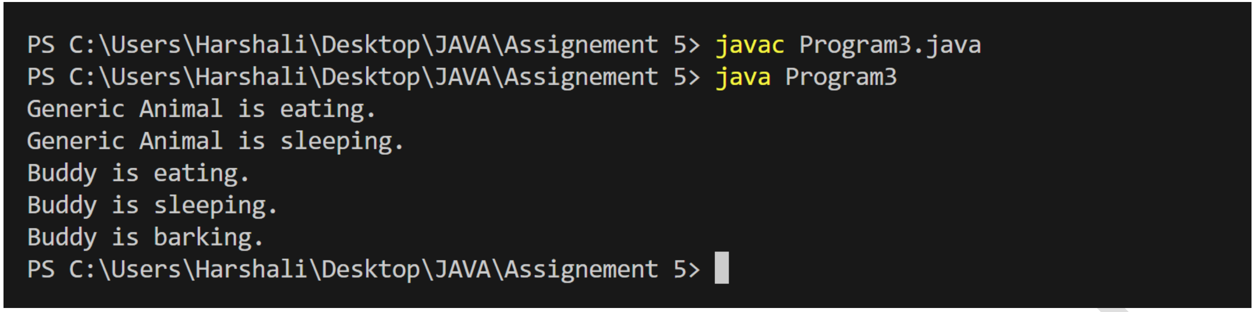
3) Create a base class Animal with attributes like name, and methods like eat() and sleep(). Create a subclass Dog that inherits from Animal and has an additional method bark(). Write a program to demonstrate the use of inheritance by creating objects of Animal and Dog and calling their methods.

Code:

```
// Base class Animal
class Animal {
   String name;
   // Constructor
   public Animal(String name) {
      this.name = name;
   }
   // Methods
   public void eat() {
      System.out.println(name + " is eating.");
   }
   public void sleep() {
      System.out.println(name + " is sleeping.");
   }
}

class Dog extends Animal {

   // Constructor
   public Dog(String name) {
      super(name);
   }
   public void bark() {
      System.out.println(name + " is barking.");
   }
}

public class Program3 {
   public static void main(String[] args) {

      Animal animal = new Animal("Generic Animal");
      animal.eat();
      animal.sleep();

      Dog dog = new Dog("Buddy");
      dog.eat();
      dog.sleep();
      dog.bark();
}
```

CDAC MUMBAI

```
}
```

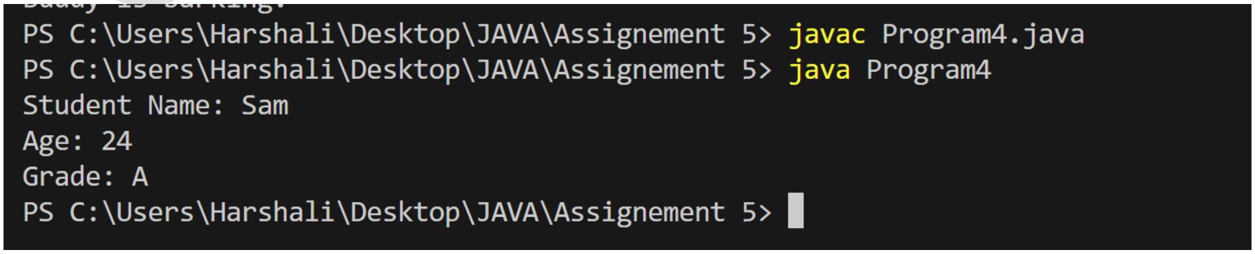


```
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5> javac Program3.java
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5> java Program3
Generic Animal is eating.
Generic Animal is sleeping.
Buddy is eating.
Buddy is sleeping.
Buddy is barking.
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5>
```

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

Code:

```
class Student {

    String name;
    int age;
    String grade;

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

    public void displayDetails() {
        System.out.println("Student Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Grade: " + grade);
    }
}

public class Program4 {
    public static void main(String[] args) {

        Student student = new Student("Sam", 24, "A");
        student.displayDetails();
    }
}
```



```
Buddy is barking.
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5> javac Program4.java
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5> java Program4
Student Name: Sam
Age: 24
Grade: A
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5>
```

CDAC MUMBAI

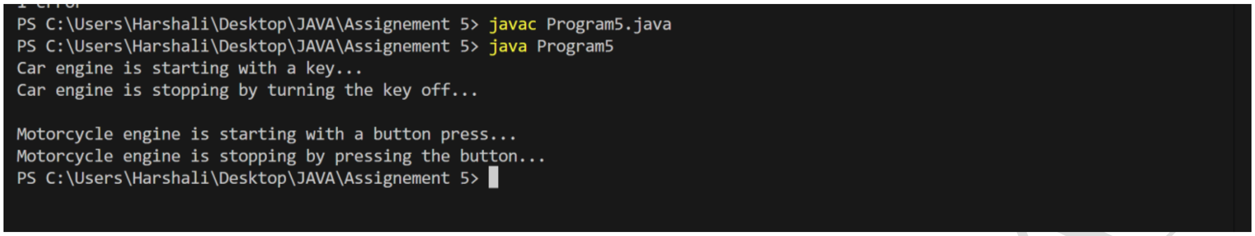
5) Write a Java program to create a base class Vehicle with methods startEngine() and stopEngine(). Create two subclasses Car and Motorcycle. Override the startEngine() and stopEngine() methods in each subclass to start and stop the engines differently.

Code:

```
class Vehicle {

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

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

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

    @Override
    public void stopEngine() {
        System.out.println("Car engine is stopping by turning the key off...");
    }
}

class Motorcycle extends Vehicle {

    @Override
    public void startEngine() {
        System.out.println("Motorcycle engine is starting with a button press...");
    }

    @Override
    public void stopEngine() {
        System.out.println("Motorcycle engine is stopping by pressing the button...");
    }
}

public class Program5 {
    public static void main(String[] args) {

        Vehicle car = new Car();
        car.startEngine();
        car.stopEngine();

        System.out.println();
```

CDAC MUMBAI

```
        Vehicle motorcycle = new Motorcycle();
        motorcycle.startEngine();
        motorcycle.stopEngine();
    }
}
```



```
1 error
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5> javac Program5.java
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5> java Program5
Car engine is starting with a key...
Car engine is stopping by turning the key off...

Motorcycle engine is starting with a button press...
Motorcycle engine is stopping by pressing the button...
PS C:\Users\Harshali\Desktop\JAVA\Assignement 5> 
```

CDAC MUMBAI