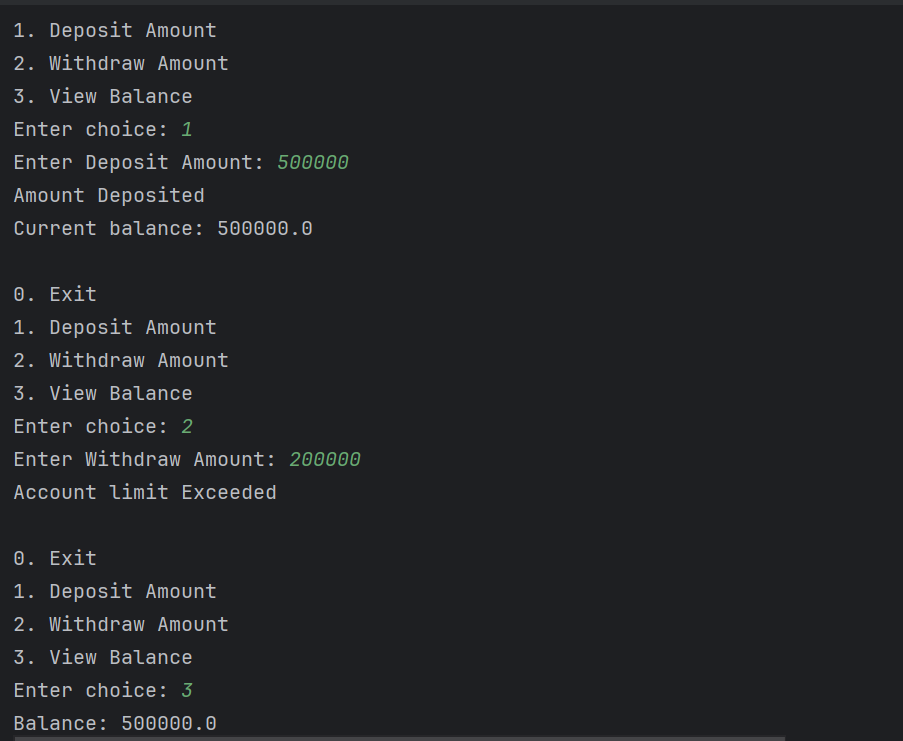
**CDAC Mumbai PG-DAC August 24**

**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.

import java.util.Scanner;  
  
class BankAccount{  
 double balance;  
  
 void deposit(double Amount){  
 this.balance+=Amount;  
 System.*out*.println("Amount Deposited");  
 System.*out*.println("Current balance: "+this.balance);  
 }  
 void withdraw(double Amount){  
 this.balance -= Amount;  
 System.*out*.println(Amount+" Rs Withdraw");  
 System.*out*.println("Current balance: "+this.balance);  
 }  
  
 void viewBalance(){  
 System.*out*.println("Balance: "+this.balance );  
 }  
}  
  
class SavingAccount extends BankAccount{  
 @Override  
 void withdraw(double Amount) {  
 if (Amount > this.balance){  
 System.*out*.println("Not Enough Bank Balance");  
 }else if (Amount > 100000){  
 System.*out*.println("Account limit Exceeded");  
 }else{  
 this.balance -= Amount;  
 System.*out*.println(Amount+" Rs Withdraw");  
 System.*out*.println("Current balance: "+this.balance);  
 }  
 }  
}  
  
  
public class Q1 {  
 static Scanner *sc* = new Scanner(System.*in*);  
 public static int menu(){  
  
 System.*out*.println("0. Exit");  
 System.*out*.println("1. Deposit Amount");  
 System.*out*.println("2. Withdraw Amount");  
 System.*out*.println("3. View Balance");  
 System.*out*.print("Enter choice: ");  
 return *sc*.nextInt();  
 }  
  
 public static void main(String[] args) {  
 BankAccount acc = new SavingAccount();  
 int choice;  
 while (( choice = *menu*()) != 0){  
 switch (choice){  
 case 1:  
 System.*out*.print("Enter Deposit Amount: ");  
 acc.deposit(*sc*.nextDouble());  
 System.*out*.println();  
 break;  
 case 2:  
 System.*out*.print("Enter Withdraw Amount: ");  
 acc.withdraw(*sc*.nextDouble());  
 System.*out*.println();  
 break;  
 case 3:  
 acc.viewBalance();  
 System.*out*.println();  
 break;  
 }  
 }  
 }  
}

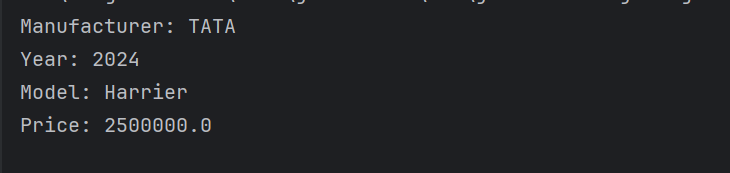
Output:



1. 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.

class Vehicle{  
 String manufacturer;  
 int Year;  
  
 Vehicle(String maker, int year){  
 this.manufacturer = maker;  
 this.Year = year;  
 }  
 void details(){}  
}  
class Car extends Vehicle{  
 String model;  
 double price;  
  
 Car(String manufacturer,int year,String model,double price){  
 super(manufacturer,year);  
 this.model = model;  
 this.price = price;  
 }  
  
 void details(){  
 System.*out*.println("Manufacturer: "+this.manufacturer);  
 System.*out*.println("Year: "+this.Year);  
 System.*out*.println("Model: "+this.model);  
 System.*out*.println("Price: "+this.price);  
 }  
  
}  
  
  
public class Q2 {  
 public static void main(String[] args) {  
 Vehicle car = new Car("TATA",2024,"Harrier",2500000);  
 car.details();  
 }  
}

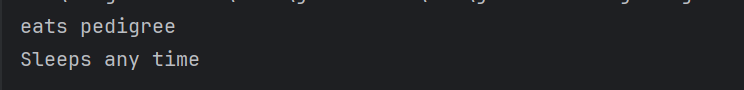
Output:



1. 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.

class Animal{  
 String name;  
  
 Animal(String name){  
 this.name = name;  
 }  
 void eats(){  
 System.*out*.println("Eats Anything");  
 }  
 void sleep(){  
 System.*out*.println("Usually sleeps at night");  
 }  
}  
class Dog extends Animal{  
 Dog(String name) {  
 super(name);  
 }  
  
 void eats(){  
 System.*out*.println("eats pedigree");  
 }  
 void sleep(){  
 System.*out*.println("Sleeps any time");  
 }  
}  
  
public class Q3 {  
 public static void main(String[] args) {  
 Animal dog = new Dog("Tom");  
 dog.eats();  
 dog.sleep();  
 }  
}

Output:

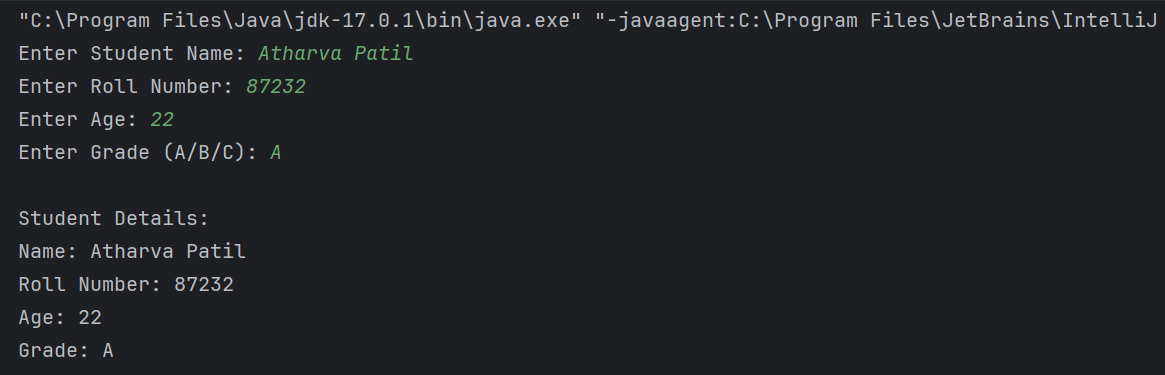


1. Build a class Student which contains details about the Student and compile and run its

instance.

import java.util.Scanner;  
  
class Student {  
 private String name;  
 private int rollNumber;  
 private int age;  
 private char grade;  
  
 void acceptDetails() {  
 Scanner sc = new Scanner(System.*in*);  
  
 System.*out*.print("Enter Student Name: ");  
 name = sc.nextLine();  
  
 System.*out*.print("Enter Roll Number: ");  
 rollNumber = sc.nextInt();  
  
 System.*out*.print("Enter Age: ");  
 age = sc.nextInt();  
  
 System.*out*.print("Enter Grade (A/B/C): ");  
 grade = sc.next().charAt(0);  
 }  
  
 void printDetails() {  
 System.*out*.println("\nStudent Details:");  
 System.*out*.println("Name: " + name);  
 System.*out*.println("Roll Number: " + rollNumber);  
 System.*out*.println("Age: " + age);  
 System.*out*.println("Grade: " + grade);  
 }  
}  
  
public class Q4 {  
 public static void main(String[] args) {  
 Student student = new Student();  
 student.acceptDetails();  
 student.printDetails();  
 }  
}

Output:



1. 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.
2. class Vehicles{  
    void startEngine(){  
    System.*out*.println("Engine started");  
    }  
    void stopEngine(){  
    System.*out*.println("Engine stopped");  
    }  
   }  
     
   class Cars extends Vehicles{  
    void startEngine(){  
    System.*out*.println("Car Engine started");  
    }  
    void stopEngine(){  
    System.*out*.println(" car Engine stopped");  
    }  
   }  
     
   class Motorcycle extends Vehicles{  
    void startEngine(){  
    System.*out*.println(" Motorcycle Engine started");  
    }  
    void stopEngine(){  
    System.*out*.println(" Motorcycle Engine stopped");  
    }  
   }  
     
   public class Q5 {  
    public static void main(String[] args) {  
    Vehicles car = new Cars();  
    car.startEngine();  
    car.stopEngine();  
    }  
   }

Output:

