**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 {

    private String accountNumber;

    private double balance;

 public BankAccount(String accountNumber, double initialBalance) {

        this.accountNumber = accountNumber;

        this.balance = initialBalance;    }

public void deposit(double amount) {

        if (amount > 0) {

            balance += amount;

            System.out.println("Deposited: " + amount);

        } else {

            System.out.println("Invalid deposit amount");

        }

    }

public boolean withdraw(double amount) {

        if (amount > 0 && amount <= balance) {

            balance -= amount;

            System.out.println("Withdrew: " + amount);

            return true;

        } else {

            System.out.println("Invalid withdraw amount or insufficient balance");

            return false;

        }

    }

public String getAccountNumber() {

        return accountNumber;

    }

public double getBalance() {

        return balance;

    }

    public String toString() {

        return "Account Number: " + accountNumber + ", Balance: " + balance;

    }

}

class SavingsAccount extends BankAccount {

    private static final double WITHDRAWAL\_LIMIT = 1000.0;

public SavingsAccount(String accountNumber, double initialBalance) {

        super(accountNumber, initialBalance);

}

    public boolean withdraw(double amount) {

        if (amount > WITHDRAWAL\_LIMIT) {

            System.out.println("Withdrawal amount exceeds the limit of " + WITHDRAWAL\_LIMIT);

            return false;

        } else {

            return super.withdraw(amount);

        }

    }

}

public class Program1 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter Savings Account Number: ");

        String accountNumber = sc.nextLine();

System.out.print("Enter Initial Balance: ");

        double initialBalance = sc.nextDouble();

        SavingsAccount savingsAccount = new SavingsAccount(accountNumber, initialBalance);

        System.out.print("Enter amount to deposit: ");

        double depositAmount = sc.nextDouble();

        savingsAccount.deposit(depositAmount);

        System.out.print("Enter amount to withdraw: ");

        double withdrawAmount = sc.nextDouble();

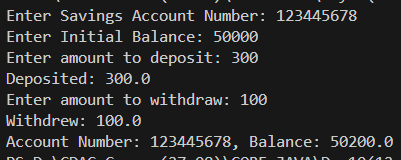
        savingsAccount.withdraw(withdrawAmount);

        System.out.println(savingsAccount);

        sc.close();

    }

}



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{

    private String make;

private int year;

  public Vehicle(String make,int year) {

        this.make=make;

      this.year=year;

    }

    public String getMake() {

        return make;

    }

    public int getYear() {

        return year;

    }

}

class Car extends Vehicle{

    String model;

    public Car(String make,int year,String model) {

        super(make, year);

        this.model=model;

    }

public String getModel() {

        return model;

    }

    public void display()

    {

        System.out.println("Make:"+getMake());

        System.out.println("Make:"+getYear());

        System.out.println("Make:"+getModel());

    }

    }

public class Program2 {

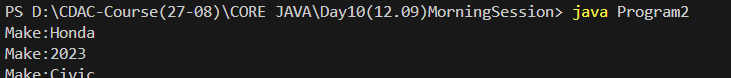
public static void main(String[] args) {

        Car c=new Car("Honda",2023,"Civic");

        c.display();

    }

}



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{

    private String name;

    public Animal(String name){

        this.name=name;

    }

    public void eat() {

        System.out.println(name+" can eat");

    }

    public void sleep() {

        System.out.println(name+" can sleep");

    }

    public String getName() {

        return name;

    }

}

class Dog extends Animal{

    public Dog(String name) {

        super(name);

    }

    public void bark() {

        System.out.println(getName()+" can bark");

    }

}

public class Program3 {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        Animal a=new Animal("Tiger");

       a.eat();

       a.sleep();

       Dog d=new Dog("Lion");

       d.eat();

       d.bark();

       d.sleep();

     }

}



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

instance.

class Student{

    private String name;

    private int rollNo;

    public Student(String name,int rollNo) {

        this.name=name;

        this.rollNo=rollNo;

    }

    public String toString() {

        return "Name:"+name+"    "+"Roll no:"+rollNo;

    }

}

public class Program4 {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

       Student s1=new Student("Ram",21);

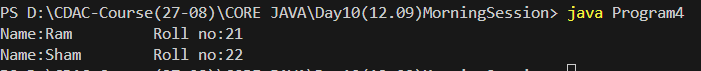
       Student s2=new Student("Sham",22);

       System.out.println(s1.toString());

       System.out.println(s2.toString());

}

}



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.

class Vehicle{

    public void startEngine() {

        System.out.println("Vehicle Engine Started");

    }

    public void stopEngine() {

        System.out.println("Vehicle Engine Stopped");

    }

}

class Car extends Vehicle{

    public void startEngine() {

        System.out.println("Car Engine Started");

    }

    public void stopEngine() {

        System.out.println("Car Engine Stopped");

    }

}

class Motorcycle extends Vehicle{

    public void startEngine() {

        System.out.println("Motorcycle Engine Started");

    }

    public void stopEngine() {

        System.out.println("Motorcycle Car Engine Stopped");

    }

}

public class Program5 {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        Vehicle v=new Vehicle();

        v.startEngine();

        v.stopEngine();

        Car c=new Car();

        c.startEngine();

        c.stopEngine();

        Motorcycle m=new Motorcycle();

        m.startEngine();

        m.stopEngine();

}

}

