**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.

*BankAccount.java*

package com.example.inh1;

import java.util.Scanner;

public class BankAccount {

protected static Scanner *sc* = new Scanner(System.***in***);

public void withdraw() {

System.***out***.println("Enter amount: ");

System.***out***.println("Please collect your ammount of Rs. " + *sc*.nextFloat());

}

public void deposit() {

System.***out***.println("Enter amount: ");

System.***out***.println("Your ammount of Rs. " + *sc*.nextFloat() + "deposited successfully!");

}

}

*SavingAccount.java*

package com.example.inh1;

public class SavingAccount extends BankAccount {

public void withdraw() {

float ammount;

System.***out***.println("Enter amount: ");

ammount = *sc*.nextFloat();

if(ammount > 100000) {

System.***out***.println("You cannot withdow more than RS. 100000");

}

else {

System.***out***.println("Please collect your ammount of Rs. " + ammount);

}

}

public void deposit() {

float ammount;

System.***out***.println("Enter amount: ");

ammount = *sc*.nextFloat();

if(ammount > 500000) {

System.***out***.println("You cannot deposit more than Rs. 500000");

}

else {

System.***out***.println("Your ammount of Rs. " + ammount + " deposited successfully!");

}

}

public void releaseResource() {

*sc*.close();

}

}

*BankService.java*

package com.example.inh1;

public class BankService {

public static void main(String[] args) {

SavingAccount sa = new SavingAccount();

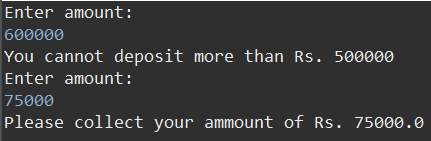
sa.deposit();

sa.withdraw();

sa.releaseResource();

}

}



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.

*Vehical.java*

package com.example.inh2;

public class Vehical {

String make;

int year;

public Vehical() {

this("",0);

}

public Vehical(String make, int year) {

this.make = make;

this.year = year;

}

}

*Car.java*

package com.example.inh2;

public class Car extends Vehical {

String model;

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

super(make, year);

this.model = model;

}

public String toString() {

return "Make: " + this.make + " Year: " + this.year + " Model: " + this.model;

}

}

*VehicalService.java*

package com.example.inh2;

import java.util.Scanner;

class VehicalService {

public static void main(String[] args) {

String make;

int year;

String model;

Scanner sc = new Scanner(System.***in***);

System.***out***.print("Enter make: ");

make = sc.next();

System.***out***.print("Enter year: ");

year = sc.nextInt();

System.***out***.print("Enter model name: ");

model = sc.next();

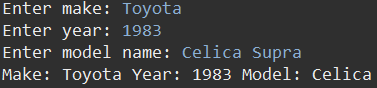
Car car = new Car(make, year, model);

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

sc.close();

}

}



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.

*Animal.java*

package com.example.inh3;

public class Animal {

public void eat() {

System.***out***.println("Animal eats food.");

}

public void sleep() {

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

}

}

*Dog.java*

package com.example.inh3;

public class Dog extends Animal{

*@Override*

public void eat() {

super.eat();

System.***out***.println("And dog likes pedigree to eat");

}

*@Override*

public void sleep() {

super.sleep();

System.***out***.println("And sleeps outside");

}

public void bark() {

System.***out***.println("Dog barks on outsiders");

}

}

*IntroducingAnimal.java*

package com.example.inh3;

public class IntroducingAnimal {

public static void main(String[] args) {

Animal an = new Dog(); //Upcasting

an.eat();

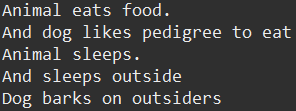
an.sleep();

Dog dog = (Dog) an; //Downcasting

dog.bark();

}

}



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

instance.

*Student.java*

package com.example.inh4;

public class Student {

String name;

long prn;

int standard;

String classroom;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public long getPrn() {

return prn;

}

public void setPrn(long prn) {

this.prn = prn;

}

public int getStandard() {

return standard;

}

public void setStandard(int standard) {

this.standard = standard;

}

public String getClassroom() {

return classroom;

}

public void setClassroom(String classroom) {

this.classroom = classroom;

}

public String toString() {

return "Name: " + this.getName() + ", Standard: " + this.getStandard() + ", Classroom: "+ this.getClassroom() + ", PRN No.: " + this.getPrn();

}

}

*StudentUtils.java*

package com.example.inh4;

import java.util.Scanner;

public class StudentUtils {

private static Scanner *sc* = new Scanner(System.***in***);

Student st = new Student();

public void acceptRecord() {

System.***out***.print("Enter name of student: ");

st.setName(*sc*.next());

System.***out***.print("Enter standard of student: ");

st.setStandard(*sc*.nextInt());

System.***out***.print("Enter classroom of student: ");

st.setClassroom(*sc*.next());

System.***out***.print("Enter PRN of student: ");

st.setPrn(*sc*.nextLong());

}

public void showRecord() {

System.***err***.println(st.toString());

}

}

*StudentRecords.java*

package com.example.inh4;

public class StudentRecords {

public static void main(String[] args) {

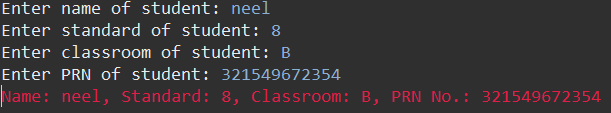
StudentUtils su = new StudentUtils();

su.acceptRecord();

su.showRecord();

}

}



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.

*Vehical.java*

package com.example.inh5;

public abstract class Vehical {

abstract public void startEngine();

abstract public void stopEngine();

}

*Car.java*

package com.example.inh5;

public class Car extends Vehical {

*@Override*

public void startEngine() {

System.***out***.println("Starting Car Engine!");

}

*@Override*

public void stopEngine() {

System.***out***.println("Stopping Car Engine!");

}

}

*Motorcycle.java*

package com.example.inh5;

public class Motorcycle extends Vehical {

*@Override*

public void startEngine() {

System.***out***.println("Starting Motorcycle Engine!");

}

*@Override*

public void stopEngine() {

System.***out***.println("Stopping Motorcycle Engine!");

}

}

*VehicalService.java*

package com.example.inh5;

class VehicalService {

public static void main(String[] args) {

Vehical veh1 = new Car(); //Upcasting

veh1.startEngine();

veh1.stopEngine();

Vehical veh2 = new Motorcycle(); //Upcasting

veh2.startEngine();

veh2.stopEngine();

}

}

