**CDAC Mumbai PG-DAC August 24**

**Assignment No- 5**

**Name:** Yash Bandu Dhole

**Centre:** Juhu.

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 problem1;

class bankAccount{

protected double balance;

public bankAccount( double initialBalance) {

this.balance = initialBalance;

}

public void deposit(double amount) {

if(amount > 0) {

balance += amount;

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

}else {

System.***out***.println("Invalid deposite amount");

}

}

public void withdraw(double amount) {

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

balance -= amount;

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

}else {

System.***out***.println("Insufficient balance or invalid amount");

}

}

public double getBalance() {

return balance;

}

public String toString() {

return "BankAccount balance " + balance;

}

}

class SavingAccount extends bankAccount{

private double withdrawalLimit;

public SavingAccount(double initialBalance, double limit) {

super(initialBalance);

this.withdrawalLimit=limit;

}

public void withdraw(double amount) {

if(amount > withdrawalLimit) {

System.***out***.println("Withdrawal amount exceeds the limit of: " + withdrawalLimit);

} else if (amount > balance){

System.***out***.println("Insufficient balance");

}else {

balance -= amount;

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

}

}

public String toString() {

return "SavingAccount balance: " + balance + ", Withdrawal Limit: " + withdrawalLimit;

}

}

public class bankOperations {

public static void main(String[] args) {

bankAccount account = new bankAccount(1000);

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

account.deposit(500);

account.withdraw(200);

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

SavingAccount savings = new SavingAccount(1500,500);

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

savings.deposit(300);

savings.withdraw(600);

savings.withdraw(400);

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

}

}

**o/p:**

BankAccount balance 1000.0

Deposited: 500.0

Withdrew: 200.0

BankAccount balance 1300.0

SavingAccount balance: 1500.0, Withdrawal Limit: 500.0

Deposited: 300.0

Withdrawal amount exceeds the limit of: 500.0

Withdrew: 400.0

SavingAccount balance: 1400.0, Withdrawal Limit: 500.0

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.

**Code:**

package problem2;

class Vehicle{

private String make;

private int year;

public Vehicle(String make, int year) {

this.make=make;

this.year= year;

}

public void displayDetails() {

System.***out***.println("Make: " +make);

System.***out***.println("Year: " + year);

}

}

class Car extends Vehicle{

protected String model;

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

super(make,year);

this.model =model;

}

public void displayDetails() {

super.displayDetails();

System.***out***.println("Model: "+ model);

}

}

public class VehicleOperation {

public static void main(String[] args) {

Car cr = new Car("Toyota", 2020, "Fortuner");

cr.displayDetails();

}

}

**O/p:**

Make: Toyota

Year: 2020

Model: Fortuner

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.

**Code:**

package problem3;

class Animal{

String name;

public Animal(String name) {

this.name=name;

}

public void eat() {

System.***out***.println(name + "is eating");

}

public void sleep() {

System.***out***.println(name + "is sleeping");

}

}

class Dog extends Animal{

public Dog (String name) {

super(name);

}

public void bark() {

System.***out***.println(name + "is barking");

}

}

public class AnimalOperations {

public static void main(String[] args) {

Animal animal = new Animal("Generic Animal");

animal.eat();

animal.sleep();

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

Dog dog = new Dog("Rocky ");

dog.eat();

dog.sleep();

dog.bark();

}

}

**O/p:**

Generic Animalis eating

Generic Animalis sleeping

Rocky is eating

Rocky is sleeping

Rocky is barking

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

instance.

**Code:**

package problem4;

class Student{

private String name;

private int rollNumber;

private char grade;

public Student(String name, int rollNumber, char grade) {

this.name= name;

this.rollNumber =rollNumber;

this.grade=grade;

}

public void displayDetails() {

System.***out***.println("Student Name: " + name);

System.***out***.println("Roll Number: "+ rollNumber);

System.***out***.println("Grade: "+ grade);

}

}

public class StudentOperations {

public static void main(String[] args) {

Student stud =new Student("Yash Dhole",101, 'A');

stud.displayDetails();

}

}

**O/p:**

Student Name: Yash Dhole

Roll Number: 101

Grade: A

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.

**Code:**

package problem5;

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{

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 VehicleOperations {

public static void main(String[] args) {

Vehicle cr=new Car();

cr.startEngine();

cr.stopEngine();

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

Vehicle bike =new Motorcycle();

bike.startEngine();

bike.stopEngine();

}

}

**O/p:**

Car engine is Starting...

Car engine is stopping...

Motorcycle engine is Starting..

Motorcycle engine is Stopping...