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 in.cdac;

public class BankAccount {

private String accHoldersName;

private double balance;

public BankAccount(){

this.accHoldersName = "";

}

public BankAccount(String accHoldersName, double balance){

this.accHoldersName = accHoldersName;

this.balance = balance;

}

public void setAccHoldersName(String accHoldersName) {

this.accHoldersName = accHoldersName;

}

public String getAccHoldersName() {

return this.accHoldersName;

}

public double getBalance() {

return this.balance;

}

public void setBalance(double balance) {

this.balance = balance;

}

public void deposit(double depositAmmount) {

this.balance = this.balance + depositAmmount;

System.out.println(depositAmmount+" RS has been deposited");

}

public void withdraw(double withdrawAmmount) {

if((withdrawAmmount <= this.balance) && (this.balance > 0) && (withdrawAmmount>0)) {

System.out.println(withdrawAmmount+" RS has been withdrawn");

this.balance = this.balance - withdrawAmmount;

}

else {

System.out.println("Invalid withdrawal ammount");

}

}

}

**SavingsAccount.java**

package in.cdac;

public class SavingsAccount extends BankAccount{

public SavingsAccount(String name, double balance) {

super(name,balance);

}

@Override

public void withdraw(double withdrawAmmount) { //Put restriction of 10,000 RS

if((withdrawAmmount <= this.getBalance()) && (this.getBalance() > 0) && (withdrawAmmount>0) && (withdrawAmmount <= 10000))

{

System.out.println(withdrawAmmount+" RS has been withdrawn");

this.setBalance(this.getBalance() - withdrawAmmount);

}

else

System.out.println("Invalid withdrawal ammount");

}

}

**Program.java**

package in.cdac;

import java.util.Scanner;

public class Program {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

SavingsAccount c1 = new SavingsAccount("Saurabh", 0);

int choice=0;

do {

System.out.println("0. Exit");

System.out.println("1. Withdraw");

System.out.println("2. Deposit");

System.out.println("3. Display account details");

System.out.print("Enter choice: ");

choice = sc.nextInt();

System.out.println();

switch(choice) {

case 1:

System.out.print("Enter withdrawal amount: ");

c1.withdraw(sc.nextDouble());

break;

case 2:

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

c1.deposit(sc.nextDouble());

break;

case 3:

System.out.println("Name: "+c1.getAccHoldersName());

System.out.println("Balance: "+c1.getBalance()+" Rs");

break;

case 0:

break;

default:

System.out.println("Invalid Choice");

break;

}

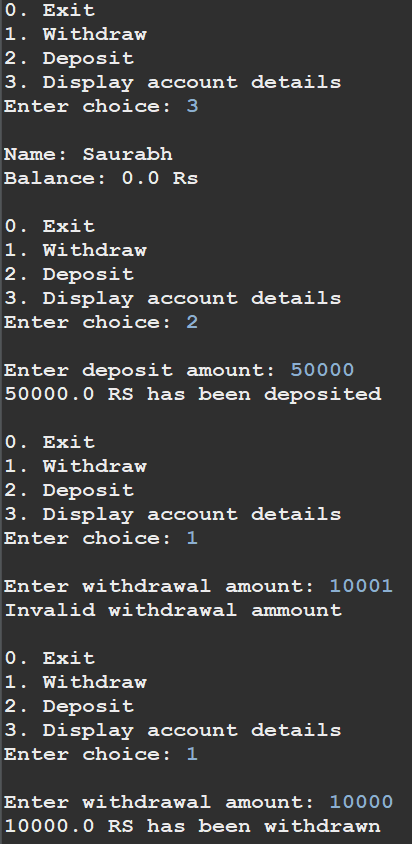
System.out.println();

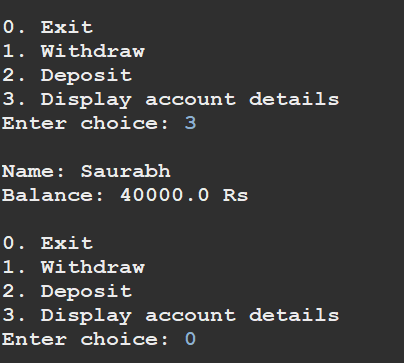
}while(choice != 0);

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.

**Vehicle.java**

package in.cdac;

public class Vehicle {

private String make;

private int year;

public Vehicle(){

this.make ="";

}

public Vehicle(String make, int year){

this.make =make;

this.year = year;

}

public String toString(){

return this.make + " " + this.year;

}

}

**Car.java**

package in.cdac;

public class Car extends Vehicle{

private String model;

public Car() {

this.model = "";

}

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

super(make,year);

this.model = model;

}

public String toString() {

return super.toString()+ " "+ this.model;

}

}

**Main.java**

package in.cdac;

public class Main {

public static void main(String[] args) {

Car c1 = new Car("SUV",2022,"Tata");

System.out.println(c1.toString());

}

}



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 in.cdac;

public class Animal {

private String name;

public Animal() {

this("");

}

public Animal(String name) {

this.name = name;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public void eat() {

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

}

public void sleep() {

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

}

}

**Dog.java**

package in.cdac;

public class Dog extends Animal {

public Dog() {

}

public Dog(String name) {

super(name);

}

public void eat() {

System.out.println(this.getName() + " is eating dog food.");

}

public void bark() {

System.out.println(this.getName()+ " is barking.");

}

}

**Main.java**

package in.cdac;

public class Main {

public static void main(String[] args) {

Animal a1 = new Animal("Lucky");

Dog d1 = new Dog("Ace");

a1.eat();

d1.eat();

d1.bark();

Animal a2 = new Dog("Buddy");

a2.eat();

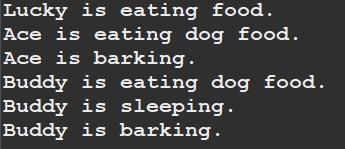
a2.sleep();

Dog d2 = (Dog)a2;

d2.bark();

}

}



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

instance.

**Student.java**

package in.cdac;

class Date{

private int day;

private int month;

private int year;

public Date() {

}

public Date(int day, int month, int year) {

this.day = day;

this.month = month;

this.year = year;

}

public int getDay() {

return day;

}

public void setDay(int day) {

this.day = day;

}

public int getMonth() {

return month;

}

public void setMonth(int month) {

this.month = month;

}

public int getYear() {

return year;

}

public void setYear(int year) {

this.year = year;

}

}

public class Student {

private String name;

private int rollNo;

private Date birthDate;

public Student() {

}

public Student(String name, int rollNo, int day, int month, int year) {

this.name = name;

this.rollNo = rollNo;

this.birthDate = new Date(day, month, year);

}

public String toString() {

return this.name + " " + this.rollNo + " " + this.birthDate.getDay() + " " + this.birthDate.getMonth() + " " + this.birthDate.getYear() + " ";

}

}

**Main.java**

package in.cdac;

public class Main {

public static void main(String[] args) {

Student s1 = new Student("Saurabh",123,19,9,1999);

System.out.println(s1.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.

**Vehicle.java**

package in.cdac;

public abstract class Vehicle {

protected String modelName;

public Vehicle() {

}

public Vehicle(String modelName) {

this.modelName = modelName;

}

abstract public void startEngine();

abstract public void stopEngine();

}

**Car.java**

package in.cdac;

public class Car extends Vehicle{

public Car() {

}

public Car(String modelName) {

super(modelName);

}

public void startEngine() {

System.out.println("Car "+this.modelName+"'s engine has been started.");

}

public void stopEngine() {

System.out.println("Car "+this.modelName+"'s engine has been stopped.");

}

}

**Motorcycle.java**

package in.cdac;

public class Motorcycle extends Vehicle{

public Motorcycle() {

}

public Motorcycle(String modelName) {

super(modelName);

}

public void startEngine() {

System.out.println("Motorcycle " +this.modelName+"'s engine has been started.");

}

public void stopEngine() {

System.out.println("Motorcycle " +this.modelName+"'s engine has been stopped.");

}

}

**Main.java**

package in.cdac;

public class Main {

public static void main(String[] args) {

Vehicle v1 = new Car("Tata Nexon");

Vehicle v2 = new Motorcycle("Pulsar");

v1.startEngine();

v1.stopEngine();

System.out.println();

v2.startEngine();

v2.stopEngine();

}

}

