An1:

import java.util.\*;

class Bank {

    private ArrayList<Account> A=new ArrayList<>();

    public void addAccount(Account a){

        A.add(a);

    }

    public Account searchAcc(double accno){

        for(int i=0;i<A.size();i++){

            if(A.get(i).getAccountNumber()==accno){

                return A.get(i);

            }

        }

        return new Account(0);

    }

}

class Account

{

    private double bal;  //The current balance

    private int accnum;  //The account number

    public Account(int a)

    {

    bal=0.0;

    accnum=a;

    }

    public void deposit(double sum)

    {

    if (sum>0)

        bal+=sum;

    else

        System.err.println("Account.deposit(...): "

                   +"cannot deposit negative amount.");

    }

    public void withdraw(double sum)

    {

    if (sum>0)

        bal-=sum;

    else

        System.err.println("Account.withdraw(...): "

                   +"cannot withdraw negative amount.");

    }

    public double getBalance()

    {

    return bal;

    }

    public double getAccountNumber()

    {

    return accnum;

    }

    public String toString()

    {

    return "Acc " + accnum + ": " + "balance = " + bal;

    }

    public final void print()

    {

    //Don't override this,

    //override the toString method

    System.out.println( toString() );

    }

}

class CurrentAccount extends Account {

    private double overdraftLimit;

    public CurrentAccount(int val,double odl){

        super(val);

        this.overdraftLimit=odl;

    }

    @Override

    public void withdraw(double sum){

        if(overdraftLimit+ getBalance()>sum) super.withdraw(sum);

        else System.out.println("Overdraft Limit exceeeded: Cannot withdraw");

    }

}

class SavingsAccount extends Account {

    private double interest;

    public SavingsAccount(int val,double interest){

        super(val);

        this.interest=interest;

    }

    public void addInterest(){

        deposit(this.getBalance()\*(this.interest));

    }

    @Override

    public void withdraw(double sum){

        if (sum<=getBalance()) super.withdraw(sum);

        else System.out.println("Cannot withdraw more than acc balance");

    }

}

public class RunBank {

    public static void main(String[] args) {

        Bank B1=new Bank();

        SavingsAccount S1=new SavingsAccount(1001,.08);

        B1.addAccount(S1);

        CurrentAccount C1=new CurrentAccount(1002,8000);

        B1.addAccount(C1);

        B1.searchAcc(1001).deposit(7000);

        B1.searchAcc(1002).deposit(10000);

        SavingsAccount S=(SavingsAccount) B1.searchAcc(1001);

        S.addInterest();

        B1.searchAcc(1001).withdraw(2000);

        B1.searchAcc(1001).print();

        B1.searchAcc(1002).withdraw(150000);

        B1.searchAcc(1002).print();

        B1.searchAcc(1002).withdraw(12500);

        B1.searchAcc(1002).print();

    }

}

An2:

class ABCIcecream{

    void icecream(){

        return;

    }

}

class XIcecream extends ABCIcecream{

    void icecream(){

        System.out.println("The icecream has vanilla flavour");

    }

}

class YIcecream extends ABCIcecream{

    void icecream(){

        System.out.println("The icecream has vanilla and chocolate flavour");

    }

}

public class GetIcecream{

    public static void main(String[] args){

        XIcecream x = new XIcecream();

        YIcecream y = new YIcecream();

        x.icecream();

        y.icecream();

    }

}

An3:

class Music{

    String songName;

    String singerName;

    Music(String songName, String singerName){

        this.songName = songName;

        this.singerName = singerName;

    }

    public void allMethods(){

        System.out.println("All methods of music class applicable");

    }

    public void play(){

        System.out.println("Song is being played from a third-party music library");

    }

}

class Sony extends Music{

    Sony(String songName, String singerName){

        super(songName, singerName);

    }

    public void play(){

        System.out.println("Song is being played from original library");

    }

}

class Panasonic extends Music{

    Panasonic(String songName, String singerName){

        super(songName, singerName);

    }

    public void play(){

    super.play();

    }

}

public class RunMusic{

    public static void main(String[] args){

        Sony s = new Sony("Kaagazi", "Kartik Sachdeva");

        Panasonic p = new Panasonic("Kun Faya Kun", "Javed Ali");

        System.out.println("Sony: ");

        s.play();

        System.out.println("Panasonic: ");

        p.play();

        p.allMethods();

    }

}

Ans4:

interface Scene{

    void display();

}

class House implements Scene{

    public void display(){

        System.out.println("House overriding scene interface method");

    }

}

class BusStop implements Scene{

    public void display(){

        System.out.println("Bus stop overriding scene interface method");

    }

}

class SupperMarket implements Scene{

    public void display(){

        System.out.println("Supper market overriding scene interface method");

    }

}

public class StartGame {

    public static void main(String[] args){

        House h = new House();

        h.display();

        BusStop b =  new BusStop();

        b.display();

        SupperMarket s = new SupperMarket();

        s.display();

    }

}