AIM:	Solve problems on Abstract
Program 1	
PROBLEM STATEMENT :	Write a program to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B.

PROGRAM:	abstract class Marks{
	abstract void getPercentage();
	}
	class A extends Marks{
	int m1,m2,m3;
	A(int m1,int m2,int m3){
	this.m1=m1;
	this.m2=m2;
	this.m3=m3;
	}
	void getPercentage(){
	System.out.println("Percentage of A: "+((m1+m2+m3)/3));
	}
	}
	class B extends Marks{
	int m1,m2,m3,m4;
	B(int m1,int m2,int m3,int m4){
	this.m1=m1;
	this.m2=m2;
	this.m3=m3;
	this.m4=m4;
	}
	void getPercentage(){
	System.out.println("Percentage of B: "+((m1+m2+m3+m4)/4));
	}
	}
	public class exp7_1 {
	public static void main(String[] args) {
	A p1 =new A(60,70,80);
	B p2=new B(60,70,80,90);
	p1.getPercentage();
	p2.getPercentage();

```
    }
}
```

RESULT:

```
Percentage of A: 70
Percentage of B: 75
```

Program 2

PROBLEM STATEMENT :

Define a class Westside that have sales in cloths and accessories. Let us say 10 cloths and 10 accessories each cost is 5000. Cloths and accessories are limited and updated as soon as purchase is done.
Define a class named Payment(abstract class) that contains an instance variable of type double that stores the amount of the payment. Amount is initialized 25,000 and updated with each purchase. Also create a method named (abstract) paymentDetails that updates the amount of the payment.
Next, define a class named CashPayment that is derived from Payment. This class should redefine the paymentDetails method to indicate that the payment is in cash. Include appropriate constructor(s)/methods.
Define a class named CreditCardPayment that is derived from Payment. This class should contain instance variables for the name on the card, expiration date, and credit card number. Include appropriate constructor(s)/methods. Finally, redefine the paymentDetails method to include all credit card information in the printout.
Define a class Person that contains person_name and P_id.
Create a main method that creates at least five persons who will be given random chance for buying using any way of payment method CashPayment /CreditCardPayment. Once person buys cloths/ accessories amount get debited.

PROGRAM:

```
import java.util.*;
class Westside{
   static int clothes = 10;
   static int accessories = 10;
   void reduce_clothes(int n){
      clothes = clothes-n;
   }
   void reduce_accessories(int m){
      accessories = accessories - m;
   }
}
abstract class Payment{
   abstract void payment_details();
}
class cashpayment extends Payment{
   static int balance;
   cashpayment(){
       balance =25000;
   }
```

```
    void reduce_cash(int n){
        balance = balance - n;
    }
    void payment_details(){
        System.out.println("Amount $"+(25000-balance) +" Paid in cash, balance :
"+this.balance);
    }
    int cash_balance(){
        return balance;
    }
}
class CreditCardPayment extends Payment{
    String cardname;
    long cardnumber;
    static int balance;
    CreditCardPayment(){
        balance = 25000;
    }
    void get_details(){
        Scanner sc=  new Scanner(System.in);
        System.out.print("Card name: ");
        cardname = sc.next();
        System.out.print("Card number: ");
        cardnumber = sc.nextLong();
    }
    void reduce_card(int n){
        balance = balance - n;
    }
    void payment_details(){
        System.out.println("Amount $"+(25000-balance)+" paid by credit card , Card Name:
"+this.cardname+" Credit Card number: "+this.cardnumber+", balance remaining
:"+this.balance);

    }
    int card_balance(){
        return balance;
    }
}
class Person{
    String name;
    int id;
    void setPerson(){
        Scanner sc=  new Scanner(System.in);
        System.out.print("Enter your name: ");
```

```
        name = sc.nextLine();
        System.out.print("Enter your id no.: ");
        id = sc.nextInt();
    }
    void getperson_details(){
        System.out.println("Name : "+this.name+" , ID : "+this.id);
    }
}
public class Main{
    public static void main(String[] args) {

        Scanner sc=  new Scanner(System.in);

        Westside w = new Westside();
        Person[] p = new Person[5];
        CreditCardPayment[] card = new CreditCardPayment[5];
        cashpayment[] cash = new cashpayment[5];

        int[] check = new int[4];
        int r =0,z;
        int index;

        while(w.clothes != 0 || w.accessories != 0){
            index = (int)(Math.random()*4);
            z=0;

            for(int i =0 ;i<r;i++){
                if(check[i]==index){
                    z++;
                }
            }

            if(r==0){
                check[0]=index;
                r++;
            }

            if(z==0){
                check[r]=index;
                r++;
            }

            System.out.println("Person : "+(index+1));
            if(z==0){
```

```
            card[index] = new CreditCardPayment();
            cash[index] = new cashpayment();
            p[index] =new Person();

            p[index].setPerson();
        }
        System.out.print("Enter no. of clothes u wish to buy: ");
        int n = sc.nextInt();
        System.out.print("Enter no. of accessories u wish to buy: ");
        int m = sc.nextInt();
        int amount = ((n+m)*(5000));
        if(card[index].card_balance()>=amount || cash[index].cash_balance()>=amount) {
            w.reduce_accessories(m);
            w.reduce_clothes(n);
            System.out.printf("Total amount to be paid: $%d\n" , amount);
            System.out.print("Press 2 to pay in cash or press 1 to pay using credit card: ");
            int option = sc.nextInt();
            if (option == 1 && card[index].card_balance()>=amount) {
                System.out.println("Enter Card Details: ");
                card[index].get_details();
                card[index].reduce_card(amount);

System.out.println("==========================================================="
);
                p[index].getperson_details();
                card[index].payment_details();

System.out.println("==========================================================="
);
            }
            else if (option == 2 && cash[index].cash_balance()>=amount) {
                cash[index].reduce_cash(amount);

System.out.println("==========================================================="
);
                p[index].getperson_details();
                cash[index].payment_details();

System.out.println("==========================================================="
);
            }
            else
                System.out.println("--------------------Transaction failed insufficient Balance------
--------------");
```

```
        }
        else
            System.out.println("--------------------Transaction failed insufficient Balance----------
-----------");
      }
   }
}
```

RESULT:

```
Person : 4
Enter your name: rao
Enter your id no.: 2334
Enter no. of clothes u wish to buy: 1
Enter no. of accessories u wish to buy: 2
Total amount to be paid: $15000
Press 2 to pay in cash or press 1 to pay using credit card: 2
================================================
Name : rao , ID : 2334
Amount $15000 Paid in cash, balance : 10000
================================================
Person : 1
Enter your name: Pranay
Enter your id no.: 2243
Enter no. of clothes u wish to buy: 2
Enter no. of accessories u wish to buy: 2
Total amount to be paid: $20000
Press 2 to pay in cash or press 1 to pay using credit card: 1
Enter Card Details:
Card name: visa
Card number: 324284234923
================================================
Name : Pranay , ID : 2243
Amount $20000 paid by credit card , Card Name: visa Credit Card number: 324284234923, balance remaining :5000
================================================
Person : 2
Enter your name: ram
Enter your id no.: 2343
Enter no. of clothes u wish to buy: 3
Enter no. of accessories u wish to buy: 0
Total amount to be paid: $15000
Press 2 to pay in cash or press 1 to pay using credit card: 2
================================================
Name : ram , ID : 2343
Amount $15000 Paid in cash, balance : 10000
================================================
Person : 1
Enter no. of clothes u wish to buy: 3
Enter no. of accessories u wish to buy: 0
Total amount to be paid: $15000
Press 2 to pay in cash or press 1 to pay using credit card: 2
--------------------Transaction failed insufficient Balance--------------------
Person : 2
Enter no. of clothes u wish to buy: 4
```

Program 3

PROBLEM STATEMENT:

There is an abstract class called 'AmusementPark'.

It has a methods

getCost() - Suppose a group comes together so this will return the total cost of the group.

There are 2 classes 'Esselworld' and 'Imagica' that inherit 'AmusementPark'. In EsselWorld, those above 21 age have an entry fee of Rs. 1050 and for ages below 21 or above 60 it's Rs. 660.

Similarly in Imagica, those above 21 age have an entry fee of Rs. 1500 and for ages below 21 or above 60 it's Rs. 1100.

Given:-

- **Esselworld has a total of 15 games**
- **Imagica has a total of 20 games**
- **Both have some games which are not included in the pass and have some**

	additional cost. (Note: You can assume games are numbered like Game 1 to 15 out of which 7 first 7 games(Game no 1 to 7) are included in the pass and the rest of the games (game no 8 to 15) have an extra charge of Rs. 50) • **When the person is going back home, the total number of games that were played and which were not played must be shown. Also, show the count of the games played.** • **On weekends Pass cost is Rs 300 higher compared to the actual cost.** • **Also, display the total cost of the person when he is leaving.(extra game cost+ticket cost)**

PROGRAM:	import java.util.Scanner;
	
	abstract class AmusementPark {
	abstract int getCost();
	}
	
	class Imagica extends AmusementPark {
	Scanner sc = new Scanner(System.in);
	int totalCost = 0;
	int age;
	String day;
	int games[] = new int[20];
	String[] games1={"Nitro","Scream Machine","Gold Rush Express","D2 Dare Drop","I For India","Rajasaurus River Adventure","Deep Space","Mr India","Alibaba Aur Chalis Chorr","Splash Ahoy!","Magic Carousel","Loch Ness Expplorers","Tubbby Takes Off Merry Go Round","Cinema 360- Prince of the Dark Waters","Save the Pirate","Curse of Salimgarh","Bandits of Robin Hood Roller Coaster","Wrath of the Gods Show","Mambo Chai Chama","Bump It Boats Ride"};
	void getData() {
	System.out.print("Enter your age: ");
	age = sc.nextInt();
	System.out.print("Enter's today day: ");
	day = sc.next();
	int N;
	System.out.print("Enter number of games played: ");
	N = sc.nextInt();
	System.out.println("Games: ");
	for(int i=0;i<20;i++){
	System.out.println((i+1)+") "+games1[i]);
	}
	System.out.print("Enter game numbers that you played: ");
	for (int i = 0; i < N; i++) {
	int temp;

```
            temp = sc.nextInt();
            games[temp - 1]++;
        }
    }

    int getCost()
    {
        if (age >= 21 && age <= 60) {
            totalCost += 1500;
        } else {
            totalCost += 1100;
        }
        for (int i = 0; i < games.length; i++) {
            if (i >= 9 && i <= 19 && games[i] == 1) {
                totalCost += 50;
            }
        }
        if (day.equals("Sunday") || day.equals("Saturday")) {
            totalCost += 300;
        }
        return totalCost;
    }

    void gamesPlayed() {
        System.out.print("The games that you have played are: ");
        for (int i = 0; i < games.length; i++) {
            if (games[i] >= 1) {
                System.out.printf("\n%s :%d",games1[i], games[i]);
            }
        }
    }
}
class Esselword extends AmusementPark {
    Scanner sc = new Scanner(System.in);
    int totalCost = 0;
    int age;
    String day;
    int games[] = new int[15];
    String[] games1={"Tunnel Twister","Monsters in the Mist","Mirror Maze","Zipper
Dripper","Highway Cars","Crazy Cups","Slippery Sultan","Hoola Loop","Super
Telecombat","Slam Bob","Rock N Roll","Happy sky","Rio Grande Train","Mini
Telecombat","Play Port"};
    void getData() {
        System.out.print("Enter your age: ");
```

```
    age = sc.nextInt();
    System.out.print("Enter's today day: ");
    day = sc.next();
    int N;
    System.out.print("Enter number of games played: ");
    N = sc.nextInt();
    System.out.println("Games: ");
    for(int i=0;i<15;i++){
        System.out.println((i+1)+") "+games1[i]);
    }
    for(int i=0;i<15;i++){
        System.out.println((i+1)+") "+games1[i]);
    }
    System.out.print("Enter game numbers that you played: ");
    for (int i = 0; i < N; i++) {
        int temp;
        temp = sc.nextInt();
        games[temp - 1]++;
    }
}

int getCost() {
    if (age >= 21 && age <= 60) {
        totalCost += 1050;
    } else {
        totalCost += 660;
    }
    for (int i = 0; i < games.length; i++) {
        if (i >= 6 && i <= 15 && games[i] == 1) {
            totalCost += 50;
        }
    }
    if (day.equals("Sunday") || day.equals("Saturday")) {
        totalCost += 300;
    }
    return totalCost;
}

void gamesPlayed() {
    System.out.print("The games that you have played are: ");
    for (int i = 0; i < games.length; i++) {
        if (games[i] >= 1) {
            System.out.printf("\n%s :%d",games1[i], games[i]);
        }
```

```
        }
    }
}

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int choice;
        System.out.print("Enter 1 for Imagica and 2 for Esselword: ");
        choice = sc.nextInt();
        switch(choice) {
            case 1:
                Imagica objI = new Imagica();
                objI.getData();

System.out.println("==========================================================
============");
                System.out.println("||||||||||||||||Thanks You Visit Again||||||||||||||||");
                System.out.println("                    Imagica                    ");
                System.out.println("\nYour total cost for today was:      ₹" + objI.getCost());
                objI.gamesPlayed();

System.out.println("\n==========================================================
==============");
                break;
            case 2:
                Esselword objE = new Esselword();
                objE.getData();

System.out.println("==========================================================
============");
                System.out.println("||||||||||||||||Thanks You Visit Again||||||||||||||||");
                System.out.println("                  EsselWorld                  ");
                System.out.println("\nYour total cost for today was:      ₹" + objE.getCost());
                objE.gamesPlayed();

System.out.println("\n==========================================================
==============");
                break;
            default:
                System.out.print("Goli beta masti nai!!!!!!");
        }
    }
}
```

RESULT:

```
Enter 1 for Imagica and 2 for Esselword: 2
Enter your age: 23
Enter's today day: Sunday
Enter number of games played: 13
Games:
1) Tunnel Twister
2) Monsters in the Mist
3) Mirror Maze
4) Zipper Dripper
5) Highway Cars
6) Crazy Cups
7) Slippery Sultan
8) Hoola Loop
9) Super Telecombat
10) Slam Bob
11) Rock N Roll
12) Happy sky
13) Rio Grande Train
14) Mini Telecombat
15) Play Port
1) Tunnel Twister
2) Monsters in the Mist
3) Mirror Maze
4) Zipper Dripper
5) Highway Cars
6) Crazy Cups
7) Slippery Sultan
8) Hoola Loop
9) Super Telecombat
10) Slam Bob
11) Rock N Roll
12) Happy sky
13) Rio Grande Train
14) Mini Telecombat
15) Play Port
Enter game numbers that you played: 1 2 3 4 5 6 7 8 9 1 3 2 14
================================================================
||||||||||||||||Thanks You Visit Again||||||||||||||||
                    EsselWorld

Your total cost for today was:       ₹1550
The games that you have played are:
Tunnel Twister :2
Monsters in the Mist :2
Mirror Maze :2
Zipper Dripper :1
Highway Cars :1
Crazy Cups :1
Slippery Sultan :1
Hoola Loop :1
Super Telecombat :1
Mini Telecombat :1
================================================================
```

CONCLUSION:	We learned to solve problems using abstract class and abstract methods