Name: Adwait S Purao
UID: 2021300101
Batch: B2
Q2:
Write a program to maintain marks of student.

i) Student is Abstract class and it has Roll no., Name, subjectmark attributes. Showstudentdata() is abstract method. Getstudentdata() is non abstract method.

ii) ISport is an Interface, having attribute sportgracemarks=5. Showsportmark() is a method.

iii) IExServiceMan is an Interface, having attribute ExServiceMangracemarks=10. ShowExServiceManmark() is a method.

iv) Result is Class and it is inheriting Student, ISport, IExServiceMan. Totalmarks=subjectmark + sportgracemarks +ExServiceMangracemarks. Showresult() is method of Result class.

Code:

```
import java.util.*;
abstract class Student{
String name;
int roll_no;
int subMarks;
Scanner sc=new Scanner(System.in);
void GetStudentData(){
System.out.println("Enter the name
of the student");
name= sc.next();
System.out.println("Enter the roll
no. of the student");
roll_no=sc.nextInt();
System.out.println("Enter the
Subject marks of the student");
subMarks=sc.nextInt();
}
abstract void ShowStudentData();
}
```

```
interface Sport{
int sportmarks=5;
abstract void ShowsportMark();
}
interface ExServiceMan{
int ExServiceManMarks=10;
void ShowExServiceManmark();
}
class Result extends Student
implements Sport,ExServiceMan{
@Override
public void ShowExServiceManmark()
{
System.out.println("The
ExServiceManMarks are
"+ExServiceManMarks);
}

@Override
public void ShowsportMark() {
System.out.println("The sportmarks
are "+sportmarks);
}
void ShowResult(){
subMarks+=
sportmarks+ExServiceManMarks;
System.out.println("The total marks
of the student are: "+subMarks);
}
```

```
@Override
void ShowStudentData() {
System.out.println("The name of the
student is: "+this.name);
System.out.println("The roll no. of
the student is: "+this.roll_no);
System.out.println("The subject
Marks of the student are:
"+this.subMarks);
}

}
public class exp8a1 {
public static void main(String[]
args) {
Result r= new Result();
r.GetStudentData();
r.ShowExServiceManmark();
r.ShowsportMark();
r.ShowStudentData();
r.ShowResult();
}
}
```

Output:

```
(base) itlab@itlab-OptiPlex-3010:~/Desktop/JavaPrograms$ javac exp8a1.java
(base) itlab@itlab-OptiPlex-3010:~/Desktop/JavaPrograms$ java exp8a1
Enter the name of the student
Adwait
Enter the roll no. of the student
101
Enter the Subject marks of the student
56
The ExServiceManMarks are 10
The sportmarks are 5
The name of the student is: Adwait
The roll no. of the student is: 101
The subject Marks of the student are: 56
The total marks of the student are: 71
(base) itlab@itlab-OptiPlex-3010:~/Desktop/JavaPrograms$
```

Q2:
Anand and Krishna are playing a game. This game starts with two piles of n1 and n2

chips. They play alternatively.

In his/her turn a person has to remove one of the piles and split the other pile into two piles, these two new piles need not be of same size. The person who cannot make a move in his turn loses. Write a program to find the winner. Here interface Piles has SetPiles(int,int) to set number of chips in each pile.

Sample Input

3 1
9 2

Sample Output

Bob
Alice

Explanation

Test case 1:

Initially, the piles are (3, 1).
Alice takes 1 and splits 3 into (1, 2).
Bob takes 1 and splits 2 into (1, 1).
Alice cannot make her move. Thus Bob wins.

Test case 2:

Initially, the piles are (9, 2).
Alice takes 9 and splits 2 into (1, 1).
Bob cannot make his move. Thus Alice wins.

Code:

```
import java.util.*;
interface games{
    void setPiles();
}
class piles implements games{
    Scanner sc= new Scanner(System.in);
    int n1,n2;
    @Override
    public void setPiles() {
        System.out.println("Enter the first
number of piles");
        n1=sc.nextInt();
        System.out.println("Enter the second
number of piles");
        n2=sc.nextInt();
    }
    void play(){
        int k=0;
        while(n1!=0 || n2!=0){
            if(n1%2!=0 && n2%2==0){
                n1=n2/2;
                n2-=n1;
                k++;
            }
```

```
            else if(n2%2!=0 && n1%2==0){
                n2=n1/2;
                n1-=n2;
                k++;
            }
            else if((n1%2!=0 && n2%2!=0) &&
n1<n2){
                n1=n2/2;
                n2-=n1;
                k++;
            }
            else if((n1%2!=0 && n2%2!=0) &&
n1>n2){
                n2=n1/2;
                n1-=n2;
                k++;
            }
            else if((n1%2==0 && n2%2==0)&&
n2>n1){
                n1=n2/2;
                n2-=n1;
                k++;
            }
            else if ((n1%2==0 && n2%2==0)&&
n1>n2) {
                n2=n1/2;
                n1-=n2;
                k++;
            }
            if((n1==1 && n2==1)){
                if(k%2==0){
                    System.out.println("Bob has
won!");
                    break;
                }
                System.out.println("Alice has
won!");
                break;
```

```
            }
        }
    }
}
public class exp8a{
    public static void main(String[] args) {
        piles p= new piles();
        p.setPiles();
        p.play();
    }
}
```

Output:

```
"C:\Program Files\Java\jdk-18.0.1\bi
Enter the first number of piles
3
Enter the second number of piles
1
Bob has won!

Process finished with exit code 0
```

```
"C:\Program Files\Java\jdk-18.0.1\bin\j
Enter the first number of piles
9
Enter the second number of piles
2
Alice has won!

Process finished with exit code 0
```