

create an interface Playable with a method play() that takes no arguments and returns void. Create three classes Football, Volleyball, and Basketball that implement the Playable interface and override the play() method to play the respective sports.

```
interface Playable {
    void play();
}

class Football implements Playable {
    String name;
    public Football(String name){
        this.name=name;
    }
    public void play() {
        System.out.println(name+" is Playing football");
    }
}
```

Similarly, create Volleyball and Basketball classes.

Sample output:

Sadhvin is Playing football
Sanjay is Playing volleyball
Sruthi is Playing basketball

For example:

Test	Input	Result
1	Sadhvin Sanjay Sruthi	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball
2	Vijay Arun Balaji	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2
3 // Define the Playable interface
4 interface Playable {
5     // Abstract method to play the respective sport
6     void play();
7 }
8
9 // Football class implementing Playable interface
10 class Football implements Playable {
11     String name;
12
13     // Constructor
14     public Football(String name) {
15         this.name = name;
16     }
17
18     // Override the play method
19     public void play() {
20         System.out.println(name + " is Playing football");
21     }
22 }
23
24 // Volleyball class implementing Playable interface
25 class Volleyball implements Playable {
26     String name;
27
28     // Constructor
29     public Volleyball(String name) {
30         this.name = name;
31     }
32
33     // Override the play method
34     public void play() {
35         System.out.println(name + " is Playing volleyball");
36     }
37 }
38
39 // Basketball class implementing Playable interface
40 class Basketball implements Playable {
41     String name;
42
43     // Constructor
44     public Basketball(String name) {
45         this.name = name;
46     }
47
48     // Override the play method
49     public void play() {
50         System.out.println(name + " is Playing basketball");
51     }
52 }
```

	Test	Input	Expected	Got	
✓	1	Sadhvin Sanjay Sruthi	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	✓
✓	2	Vijay Arun Balaji	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	✓

Passed all tests! ✓

RBI issues all national banks to collect interest on all customer loans.

Create an RBI interface with a variable String parentBank="RBI" and abstract method rateOfInterest().

RBI interface has two more methods default and static method.

```
default void policyNote() {  
    System.out.println("RBI has a new Policy issued in 2023.");  
}  
  
static void regulations(){  
    System.out.println("RBI has updated new regulations on 2024.");  
}
```

Create two subclasses SBI and Karur which implements the RBI interface.

Provide the necessary code for the abstract method in two sub-classes.

Sample Input/Output:

RBI has a new Policy issued in 2023
RBI has updated new regulations in 2024.
SBI rate of interest: 7.6 per annum.
Karur rate of interest: 7.4 per annum.

For example:

Test	Result
1	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.

Answer: (penalty regime: 0 %)

```
1 // Define the RBI interface  
2 interface RBI {  
3     // Variable declaration  
4     String parentBank = "RBI";  
5  
6     // Abstract method  
7     double rateOfInterest();  
8  
9     // Default method  
10    default void policyNote() {  
11        System.out.println("RBI has a new Policy issued in 2023");  
12    }  
13  
14    // Static method  
15    static void regulations() {  
16        System.out.println("RBI has updated new regulations in 2024.");  
17    }  
18 }  
19  
20 // SBI class implementing RBI interface  
21 class SBI implements RBI {  
22     // Implementing the abstract method  
23     public double rateOfInterest() {  
24         return 7.6;  
25     }  
26 }  
27  
28 // Karur class implementing RBI interface  
29 class Karur implements RBI {  
30     // Implementing the abstract method  
31     public double rateOfInterest() {  
32         return 7.4;  
33     }  
34 }  
35  
36 // Main class to test the functionality  
37 public class Main {  
38     public static void main(String[] args) {  
39         // RBI policies and regulations  
40         RBI rbi = new SBI(); // Can be any class implementing RBI  
41         rbi.policyNote();    // Default method  
42         RBI.regulations();   // Static method  
43  
44         // SBI bank details  
45         SBI sbi = new SBI();  
46         System.out.println("SBI rate of interest: " + sbi.rateOfInterest() + " per annum.");  
47  
48         // Karur bank details  
49         Karur karur = new Karur();  
50         System.out.println("Karur rate of interest: " + karur.rateOfInterest() + " per annum.");  
51     }  
52 }
```

	Test	Expected	Got	
✓	1	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	✓

Passed all tests! ✓

Create interfaces shown below.

```
interface Sports {
    public void setHomeTeam(String name);
    public void setVisitingTeam(String name);
}
```

```
interface Football extends Sports {
    public void homeTeamScored(int points);
    public void visitingTeamScored(int points);}
create a class College that implements the Football interface and provides the necessary functionality to the abstract methods.
```

sample Input:

Rajalakshmi

Saveetha

22

21

Output:

Rajalakshmi 22 scored

Saveetha 21 scored

Rajalakshmi is the Winner!

For example:

Test	Input	Result
1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!

Answer: (penalty regime: 0 %)

Reset answer

```
1 import java.util.Scanner;
2
3 interface Sports {
4     void setHomeTeam(String name);
5     void setVisitingTeam(String name);
6 }
7
8 interface Football extends Sports {
9     void homeTeamScored(int points);
10    void visitingTeamScored(int points);
11 }
12
13 class College implements Football {
14     private String homeTeam;
15     private String visitingTeam;
16     private int homeTeamPoints = 0;
17     private int visitingTeamPoints = 0;
18
19     public void setHomeTeam(String name) {
20         this.homeTeam = name;
21     }
22
23     public void setVisitingTeam(String name) {
24         this.visitingTeam = name;
25     }
26
27     public void homeTeamScored(int points) {
28         homeTeamPoints += points;
29         System.out.println(homeTeam + " " + points + " scored");
30     }
31
32     public void visitingTeamScored(int points) {
33         visitingTeamPoints += points;
34         System.out.println(visitingTeam + " " + points + " scored");
35     }
36
37     public void winningTeam() {
38         if (homeTeamPoints > visitingTeamPoints) {
39             System.out.println(homeTeam + " is the winner!");
40         } else if (homeTeamPoints < visitingTeamPoints) {
41             System.out.println(visitingTeam + " is the winner!");
42         } else {
43             System.out.println("It's a tie match.");
44         }
45     }
46 }
47
48 public class Main {
49     public static void main(String[] args) {
50         Scanner sc = new Scanner(System.in);
51
52         // Get home team name
```

	Test	Input	Expected	Got	
✓	1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	✓
✓	2	Anna Balaji 21 21	Anna 21 scored Balaji 21 scored It's a tie match.	Anna 21 scored Balaji 21 scored It's a tie match.	✓
✓	3	SRM VIT 20 21	SRM 20 scored VIT 21 scored VIT is the winner!	SRM 20 scored VIT 21 scored VIT is the winner!	✓

Passed all tests! ✓

◀ Lab-07-MCQ

Jump to...

Generate series and find Nth element ▶