

[Dashboard](#) / [My courses](#) / [CS23333-OOPUI-2023](#) / [Lab-07-Interfaces](#) / [Lab-07-Logic Building](#)

<b>Status</b>	Finished
<b>Started</b>	Sunday, 6 October 2024, 10:47 AM
<b>Completed</b>	Sunday, 6 October 2024, 6:20 PM
<b>Duration</b>	7 hours 32 mins

## Question 1

Correct

Marked out of 5.00

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 interface Sports {
3     public void setHomeTeam(String name);
4     public void setVisitingTeam(String name);
5 }
6
7 interface Football extends Sports {
8     public void homeTeamScored(int points);
9     public void visitingTeamScored(int points);
10 }
11
12 class College implements Football {
13     String homeTeam;
14     String visitingTeam;
15     int homeTeamPoints=0;
16     int visitingTeamPoints=0;
17
18     public void setHomeTeam(String name){
19         this.homeTeam = name;
20     }
21     public void setVisitingTeam(String name){
22         this.visitingTeam = name;
23     }
24     public void homeTeamScored(int points){
25         homeTeamPoints+=points;
26         System.out.println(homeTeam+" "+points+" scored");
27     }
28     public void visitingTeamScored(int points){
29         visitingTeamPoints+=points;
30         System.out.println(visitingTeam+" "+points+" scored");
31     }
32     public void winningTeam(){
33         if(homeTeamPoints>visitingTeamPoints){
34             System.out.println(homeTeam+" is the winner!");
35         }
36     }
37 }
```

```

36
37 v   else if(homeTeamPoints<visitingTeamPoints){
38       System.out.println(visitingTeam+" is the winner!");
39   }
40
41 v   else{
42       System.out.println("It's a tie match.");
43   }
44
45 }
46 }
47 v class prog{
48 v     public static void main(String[] args){
49         String hname;
50         Scanner sc= new Scanner(System.in);
51         hname = sc.nextLine();
52         String vteam=sc.next();

```

	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! ✓

## Question 2

Correct

Marked out of 5.00

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 // Define the Playable interface
3 interface Playable {
4     // Abstract method to play the respective sport
5     void play();
6 }
7 // Football class implementing Playable interface
8 class Football implements Playable {
9     String name;
10    // Constructor
11    public Football(String name) {
12        this.name = name;
13    }
14    // Override the play method
15    public void play() {
16        System.out.println(name + " is Playing football");
17    }
18 }
19 // Volleyball class implementing Playable interface
20 class Volleyball implements Playable {
21     String name;
22     // Constructor
23     public Volleyball(String name) {
24         this.name = name;
25     }
26     // Override the play method
27     public void play() {
28         System.out.println(name + " is Playing volleyball");
29     }
30 }
31 // Basketball class implementing Playable interface
32 class Basketball implements Playable {
33     String name;
```

```

34 // Constructor
35 public Basketball(String name) {
36     this.name = name;
37 }
38 // Override the play method
39 public void play() {
40     System.out.println(name + " is Playing basketball");
41 }
42 }
43 // Main class to test the functionality
44 public class Main {
45     public static void main(String[] args) {
46         Scanner scanner = new Scanner(System.in);
47         // Input for Football player
48         String footballPlayerName = scanner.nextLine();
49         Football footballPlayer = new Football(footballPlayerName);
50         // Input for Volleyball player
51         String volleyballPlayerName = scanner.nextLine();
52         Volleyball volleyballPlayer = new Volleyball(volleyballPlayerName);

```

	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! ✓

## Question 3

Correct

Marked out of 5.00

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 public class Main {
36     public static void main(String[] args) {
37         // RBI policies and regulations

```

```

38     RBI rbi = new SBI(); // Can be any class implementing RBI
39     rbi.policyNote();    // Default method
40     RBI.regulations();   // Static method
41
42     // SBI bank details
43     SBI sbi = new SBI();
44     System.out.println("SBI rate of interest: " + sbi.rateOfInterest() + " per annum.");
45
46     // Karur bank details
47     Karur karur = new Karur();
48     System.out.println("Karur rate of interest: " + karur.rateOfInterest() + " per annum.");
49 }
50 }

```

	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! ✓

◀ Lab-07-MCQ

Jump to...

Generate series and find Nth element ▶