Week07 - INTERFACES

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

```
import java.util.Scanner;

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");
    }
}
class Volleyball implements Playable {
```

```
    String name;
    public Volleyball(String name) {
        this.name = name;
    }
    public void play() {
        System.out.println(name + " is Playing volleyball");
    }
}
class Basketball implements Playable {
    String name;
    public Basketball(String name) {
        this.name = name;
    }
    public void play() {
        System.out.println(name + " is Playing basketball");
    }
}
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Reading names from input dynamically
        String footballPlayerName = scanner.nextLine();
        String volleyballPlayerName = scanner.nextLine();
        String basketballPlayerName = scanner.nextLine();
        // Create players dynamically based on input
        Football footballPlayer = new Football(footballPlayerName);
        Volleyball volleyballPlayer = new Volleyball(volleyballPlayerName);
        Basketball basketballPlayer = new Basketball(basketballPlayerName);
        // Play the respective sports
        footballPlayer.play();
        volleyballPlayer.play();
```

```
        basketballPlayer.play();
        scanner.close();
    }
}
```

	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! ✓

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!

```
import java.util.Scanner;

// Sports Interface

interface Sports {

    public void setHomeTeam(String name);
```

```
    public void setVisitingTeam(String name);
}
// Football Interface extending Sports
interface Football extends Sports {
    public void homeTeamScored(int points);
    public void visitingTeamScored(int points);
}
// College class implementing Football interface
class College implements Football {
    private String homeTeam;
    private String visitingTeam;
    private int homeTeamPoints;
    private int visitingTeamPoints;
    // Implementing setHomeTeam method
    @Override
    public void setHomeTeam(String name) {
        this.homeTeam = name;
    }
    // Implementing setVisitingTeam method
    @Override
    public void setVisitingTeam(String name) {
        this.visitingTeam = name;
    }
    // Implementing homeTeamScored method
    @Override
    public void homeTeamScored(int points) {
        this.homeTeamPoints = points;
    }
    // Implementing visitingTeamScored method
    @Override
    public void visitingTeamScored(int points) {
```

```
        this.visitingTeamPoints = points;
    }
    // Method to display the result
    public void displayResult() {
        System.out.println(homeTeam + " " + homeTeamPoints + " scored");
        System.out.println(visitingTeam + " " + visitingTeamPoints + " scored");
        if (homeTeamPoints > visitingTeamPoints) {
            System.out.println(homeTeam + " is the winner!");
        } else if (visitingTeamPoints > homeTeamPoints) {
            System.out.println(visitingTeam + " is the winner!");
        } else {
            System.out.println("It's a tie match.");
        }
    }
}
// Main class to execute the program
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Reading input dynamically
        String homeTeam = scanner.nextLine();       // First input: Home team name
        String visitingTeam = scanner.nextLine();   // Second input: Visiting team name
        int homeTeamScore = scanner.nextInt();      // Third input: Home team score
        int visitingTeamScore = scanner.nextInt();  // Fourth input: Visiting team score
        // Creating an instance of College class
        College collegeMatch = new College();
        // Setting teams and scores
        collegeMatch.setHomeTeam(homeTeam);
        collegeMatch.setVisitingTeam(visitingTeam);
        collegeMatch.homeTeamScored(homeTeamScore);
        collegeMatch.visitingTeamScored(visitingTeamScore);
```

```
    // Displaying the result
    collegeMatch.displayResult();
}
```

	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! ✓

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.

```
interface RBI{
    String parentBank ="RBI";
    double rateofInterest();
```

Week07 - INTERFACES

```
    default void policyNote() {
       System.out.println("RBI has updated new regulations in 2024.");}
    static void regulations() {
        System.out.println("RBI has a new Policy issued in 2023");}
}
class SBI  implements RBI{
  public double rateofInterest() {
         return 7.6;
}
}
class Karur implements RBI {
    public double rateofInterest() {
        return 7.4;
}
}
public class Main {
 public static void main(String[] args) {
    RBI.regulations();
SBI sbiBank =new SBI();
Karur karurBank =new Karur();
sbiBank.policyNote();
System.out.println("SBI rate of interest: "+ sbiBank.rateofInterest()+" per annum.");
System.out.println("Karur rate of interest: "+karurBank.rateofInterest()+" per annum.");
}
}
```

	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! ✓