### WEEK-07

### Question 1

### 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:

### 

### Program:

### 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 Prog {

### public static void main(String[] args) {

### Scanner scan = new Scanner(System.in);

### 

### String name = scan.nextLine();

### Football foot = new Football(name);

### 

### name = scan.nextLine();

### Volleyball volley = new Volleyball(name);

### 

### name = scan.nextLine();

### Basketball basket = new Basketball(name);

### 

### // Call the play method for each player

### foot.play();

### volley.play();

### basket.play();

### 

### scan.close(); // Close the scanner

### }

### }

### 

### Question 2

### 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 | RajalakshmiSaveetha2221 | Rajalakshmi 22 scoredSaveetha 21 scoredRajalakshmi is the winner! |

### Program:

### import java.util.Scanner;

### 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);

### }

### class College implements Football {

### String homeTeam;

### String visitingTeam;

### 

### public void setHomeTeam(String name){

### this.homeTeam=name;

### 

### }

### public void setVisitingTeam(String name){

### visitingTeam=name;

### }

### public void homeTeamScored(int points){

### System.out.println(homeTeam+" "+points+" scored");

### }

### public void visitingTeamScored(int points){

### System.out.println(visitingTeam+" "+points+" scored");

### }

### public void winningTeam(int p1, int p2){

### if(p1>p2)

### System.out.println(homeTeam+" is the winner!");

### else if(p1<p2)

### System.out.println(visitingTeam+" is the winner!");

### else

### System.out.println("It's a tie match.");

### }

### }

### class prog{

### public static void main(String[] args){

### String hname;

### Scanner sc= new Scanner(System.in);

### hname=sc.nextLine();

### String vteam=sc.next();

### int htpoints=sc.nextInt();

### int vtpoints=sc.nextInt();

### College s= new College();

### s.setHomeTeam(hname);

### s.setVisitingTeam(vteam);

### s.homeTeamScored(htpoints);

### s.visitingTeamScored(vtpoints);

### s.winningTeam(htpoints,vtpoints);

### }

### }

### 

### Question 3

### 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:

### 

### Program:

### interface RBI {

### String parentBank = "RBI"; // This is a constant

### double rateOfInterest(); // Method to get the rate of interest

### 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 in 2024.");

### }

### }

### class SBI implements RBI {

### public double rateOfInterest() {

### return 7.6; // SBI rate of interest

### }

### }

### class Karur implements RBI {

### public double rateOfInterest() {

### return 7.4; // Karur rate of interest

### }

### }

### public class Main {

### public static void main(String[] args) {

### SBI sbi = new SBI(); // Corrected instantiation of SBI

### Karur karur = new Karur(); // Corrected instantiation of Karur

### sbi.policyNote(); // Calling default method

### RBI.regulations(); // Calling static method

### // Print the rates of interest

### System.out.println("SBI rate of interest: " + sbi.rateOfInterest() + " per annum.");

### System.out.println("Karur rate of interest: " + karur.rateOfInterest() + " per annum."); // Fixed concatenation

### }

### }

### 