**MODULE 2**

**SECTION 8**

1. It’s been a brutally cold and snowy winter. None of your friends have wanted to play soccer. But  
   now that spring has arrived, another season of the league can begin. Your challenge is to write a  
   program that models a soccer league and keeps track of the season’s statistics.  
   There are 4 teams in the league. Matchups are determined at random. 2 games are played every  
   Tuesday, which allows every team to participate weekly. There is no set number of games per  
   season. The season continues until winter arrives.  
   The league is very temperature-sensitive. Defenses are sluggish on hot days. Hotter days allow for  
   the possibility of more goals during a game. If the temperature is freezing, no games are played  
   that week. If there are 3 consecutive weeks of freezing temperatures, then winter has arrived and the season is over.  
   Tasks  
   Write a program that models a soccer league and keeps track of the season’s statistics. Carefully consider what data should be stored  
   in an array and what data should be stored in an ArrayList. Design classes with fields and methods based on the description of the  
   league. You’ll also need a test class that contains a main method. All fields must be private. Provide any necessary getters and  
   setters.  
   Teams  
   Each team has a name. The program should also keep track of each team’s win-total, loss-total, tie-total, total goals scored, and total  
   goals allowed. Create an array of teams that the scheduler will manage.  
   Print each team’s statistics when the season ends.  
   Games  
   In a game, it’s important to note each team’s name, each team’s score, and the temperature that day. Number each game with integer  
   ID number. This number increases as each game is played. Keep track of every game played this season. This class stores an  
   ArrayList of all games as a field.  
   Your program should determine scores at random. The maximum number of goals any one team can score should increase  
   proportionally with the temperature. But make sure these numbers are somewhat reasonable.  
   When the season ends, print the statistics of each game. Print the hottest temperature and average temperature for the season.  
   Scheduler  
   Accept user input through a JOptionPane or Scanner. While the application is running, ask the user to input a temperature. The  
   program should not crash because of user input. If it’s warm enough to play, schedule 2 games. Opponents are chosen at random.  
   Make sure teams aren’t scheduled to play against themselves. If there are 3 consecutive weeks of freezing temperatures, the season  
   is over.

**ANSWER:**

import java.util.ArrayList;

import java.util.Random;

import java.util.Scanner;

// Team class

class Team {

private String name;

private int wins;

private int losses;

private int ties;

private int goalsScored;

private int goalsAllowed;

public Team(String name) {

this.name = name;

this.wins = 0;

this.losses = 0;

this.ties = 0;

this.goalsScored = 0;

this.goalsAllowed = 0;

}

public String getName() {

return name;

}

public int getWins() {

return wins;

}

public int getLosses() {

return losses;

}

public int getTies() {

return ties;

}

public int getGoalsScored() {

return goalsScored;

}

public int getGoalsAllowed() {

return goalsAllowed;

}

public void addWin() {

this.wins++;

}

public void addLoss() {

this.losses++;

}

public void addTie() {

this.ties++;

}

public void addGoalsScored(int goals) {

this.goalsScored += goals;

}

public void addGoalsAllowed(int goals) {

this.goalsAllowed += goals;

}

}

// Game class

class Game {

private static int gameCounter = 0;

private int gameId;

private Team team1;

private Team team2;

private int team1Score;

private int team2Score;

private int temperature;

public Game(Team team1, Team team2, int temperature) {

this.gameId = ++gameCounter;

this.team1 = team1;

this.team2 = team2;

this.temperature = temperature;

this.team1Score = calculateScore(temperature);

this.team2Score = calculateScore(temperature);

}

private int calculateScore(int temperature) {

Random random = new Random();

return random.nextInt(temperature / 10 + 1); // Simplified scoring based on temperature

}

public void play() {

team1.addGoalsScored(team1Score);

team1.addGoalsAllowed(team2Score);

team2.addGoalsScored(team2Score);

team2.addGoalsAllowed(team1Score);

if (team1Score > team2Score) {

team1.addWin();

team2.addLoss();

} else if (team1Score < team2Score) {

team2.addWin();

team1.addLoss();

} else {

team1.addTie();

team2.addTie();

}

}

public String toString() {

return "Game ID: " + gameId + " | " + team1.getName() + " " + team1Score + " - " + team2Score + " " + team2.getName() + " | Temperature: " + temperature;

}

public int getTemperature() {

return temperature;

}

}

// Scheduler class

class Scheduler {

private Team[] teams;

private ArrayList<Game> games;

private int consecutiveFreezingWeeks;

public Scheduler() {

teams = new Team[] {

new Team("Team A"),

new Team("Team B"),

new Team("Team C"),

new Team("Team D")

};

games = new ArrayList<>();

consecutiveFreezingWeeks = 0;

}

public void scheduleGames(int temperature) {

if (temperature <= 32) {

consecutiveFreezingWeeks++;

System.out.println("No games played this week due to freezing temperature.");

if (consecutiveFreezingWeeks >= 3) {

System.out.println("Winter has arrived. Season is over.");

printSeasonStatistics();

System.exit(0);

}

return;

}

consecutiveFreezingWeeks = 0;

Random random = new Random();

ArrayList<Integer> usedIndexes = new ArrayList<>();

for (int i = 0; i < 2; i++) {

int team1Index = random.nextInt(4);

while (usedIndexes.contains(team1Index)) {

team1Index = random.nextInt(4);

}

usedIndexes.add(team1Index);

int team2Index = random.nextInt(4);

while (usedIndexes.contains(team2Index) || team1Index == team2Index) {

team2Index = random.nextInt(4);

}

usedIndexes.add(team2Index);

Game game = new Game(teams[team1Index], teams[team2Index], temperature);

game.play();

games.add(game);

}

}

public void printSeasonStatistics() {

System.out.println("Season Statistics:");

for (Team team : teams) {

System.out.println(team.getName() + ": " +

"Wins: " + team.getWins() + ", " +

"Losses: " + team.getLosses() + ", " +

"Ties: " + team.getTies() + ", " +

"Goals Scored: " + team.getGoalsScored() + ", " +

"Goals Allowed: " + team.getGoalsAllowed());

}

int hottestTemperature = Integer.MIN\_VALUE;

int totalTemperature = 0;

for (Game game : games) {

System.out.println(game);

if (game.getTemperature() > hottestTemperature) {

hottestTemperature = game.getTemperature();

}

totalTemperature += game.getTemperature();

}

System.out.println("Hottest Temperature: " + hottestTemperature);

System.out.println("Average Temperature: " + (games.isEmpty() ? 0 : totalTemperature / games.size()));

}

}

// Test class

public class SoccerLeague {

public static void main(String[] args) {

Scheduler scheduler = new Scheduler();

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.print("Enter the temperature for this week: ");

String input = scanner.nextLine();

try {

int temperature = Integer.parseInt(input);

scheduler.scheduleGames(temperature);

} catch (NumberFormatException e) {

System.out.println("Invalid input. Please enter a valid integer temperature.");

}

}

}

}