State Board of Cricket Council –V2.0 *

State Board of Cricket Council

State Board of Cricket Council (SBCC) is one of the leading cricket selection academies in the state. They are in need of an automated system that should manipulate the player details provided and also find the players who have secured star rating between a specific range from the database.

You being their software consultant have been approached to develop a pilot java application which can be used by the admin for the above mentioned requirement.

Click below to download Requirement Document(s)

Requirement Document - 1/7

Requirement Document - 2/7

State Board of Cricket Council – Running Case study

Requirement 2: Parse the player details and create a player profile

This functionality deals with getting the basic details of players such as player Id, player Name, number of matches played, runScored and playing zone of the player in the form of String. Create a player by parsing the input string and display as specified in the sample input and output.

The input string that has the player details will be stored in the following format.

playerId:playerName:matchesPlayed:runscoredineachmatchseperatedbycolon:playingZone:playerType:numberofhundredsormaidenovers:numberoffiftiesorhatrickwickets

For eg:

HXCB1234D:Dhoni:5:20:130:55:102:100:North:Batsman:3:1

SAFG1243P:Mahee:3:20:30:55:South:Bowler:4:0

Note: In the provided input string, if a player has played **N** matches then there will be runs scored in **N** matches. In the above example Dhoni has played 5 matches so there are runs scored in 5 matches.

The parsePlayerDetails method in the SBCCUtility class should take this input string as an argument and should parse the String and create a Player by setting the values for all the attributes of the Player and return the same.

Note: To calculate the value for the attribute runScored , this method has to invoke the calculateTotalRuns method in the Player class by passing the details of runs scored by the player in each match as an argument.

Component Specification: SBCCUtility Class

Component Name	**Type(Class)**	**Attributes**	**Methods**	**Responsibilities**
Parse the player details and create a Player	SBCCUtility		public Player parsePlayerDetails(String playerDetails)	This method takes the String which holds all the player details as argument, it should parse the data stored in the String, Invoke the calculateTotalRuns method in the player class and create the Player object and return the same.

The calculateTotalRuns method in the Player class takes a string array as an argument. This string array will have the details of runs scored by the player in each match. This method has to sum the runs scored by the player in each match and return the sum.

For eg: If the string array passed as an argument contains the following values,

Runs scored by player in 5 matches = **20, 130, 55, 102, 100**

Then, total runs can be calculated as **20 + 130 + 55 + 102 +100** ==> **407**

Component Specification: Player (Model Class)

Component Name	**Type (Class)**	**Attributes**	**Methods**	**Responsibilities**
Calculate the total runs scored by the	**Player**		**public int calculateTotalRuns(String[] securedRuns)**	**This method takes** the **String array as an argument which contains the runs**

Player				**scored by the player in each match. It has to calculate the total runs scored by the player by summing the runs scored by the player in each match and return the sum.**
	Player	String playerId String playerName int matchesPlayed int runScored String playingZone	Include all necessary Getters and Setters for all the attributes Provide a no argument and a five argument constructor in the given order playerId, playerName, matchesPlayed, runScored and playingZone.	

In the UserInterface class, **in the main method provided, fill the code to produce the output as shown in the Sample input and Output.**

When the user selects option **1 i.e., Parse the player details and create player**, it should get the player details from the user and display the player details.

When the user selects option **2 i.e., Exit**, display the message "**Thank you for using SBCC application**" and end the program.

OVERALL DESIGN CONSTRAINTS:

- The Player class should be inside the package com.sbcc.model
- **The SBCCUtility class should be inside the package com.sbcc.utility**
- The UserInterface class should be inside the package com.sbcc.main
- Adhere to the design specifications mentioned in the case study.

- **The classes and methods should be declared as public and all the attributes should be declared as private.**
- **Do not change or delete the class/method/attributes, names or return types which are provided to you as a part of the base code skeleton.**
- Please make sure that your code does not have any compilation errors while submitting.

Sample Input and Output 1 [Values given in bold represents the input]:

1. Parse the player details and create player

2. Exit

Enter your choice

1

Enter the player details

HXCB1234D:Dhoni:5:20:130:55:102:100:North:Batsman:3:1

Player id: HXCB1234D

Player name: Dhoni

Matches played: 5

Total runs scored: 407

Playing zone: North

1. Parse the player details and create player

2. Exit

Enter your choice

2

Thank you for using SBCC application