

Aayush Shah

Comprehensive Assignment

22 November 2020

Comprehensive Assignment

Object Oriented Programming

Definition :

Department of Computer Science & Engineering
Institute of Technology, Nirma University
Semester III (B.Tech CSE)
2CS302 Object Oriented Programming
Assignment for Comprehensive Evaluation

Instructions:

1. Assume suitable data /information if required.

Develop the Java program for the cricket information portal as mentioned below:

- Data/Attributes for portal:
 - playerFirstName, matchDate, runScore, wicketTaken, roleInTeam (Bowler/Batsman)
- The portal has following functionalities:
 - Add match in information
 - Display match information
 - Display player-wise maximum 3 run scored (in descending order) details.
i.e.

Player Name: Virat

matchDate	runScore
9/10/2020	65
7/10/2020	45
18/10/2020	40

..... (other player details)

- Find and Display top 3 bowler (using total run) and top 3 batsman (using total wickets). You must have same number of records for each player.
- You have to use parameterized constructor, array and exception handling concepts compulsory. You may use other concepts of Object Oriented Programming if required.

Name of file : Main.java

File purpose : To develop the cricket information portal.

Java file code :

```
1. import java.util.*;
2.
3. class WrongDate extends Exception{           //
   class which represents the use-
   defined exception when user entered invalid date.
4.     public WrongDate(String s){
5.         super(s);           //
   call constructor of parent exception
6.     }
7. }
8. class InvalidWicket extends Exception{       //
   class which represents the use-
   defined exception when user entered more than 10 wickets
   per match.
9.     public InvalidWicket(String s){
10.        super(s);           //
   call constructor of parent excpetion
11.    }
12. }
13. class Date{           //
   class which represents the date format
14.     int day,month,year;
15.     Date(){           //constructor without argument
16.         day = 8;           //assigning random value
17.         month = 12;
18.         year = 2000;
19.     }
20.     Date(int d, int m, int y){           //
   constructor with arguments
21.         day = d;
22.         month = m;
23.         year = y;
24.     }
25.
26.     void setDay(int day){           //set method for day
27.         this.day = day;
28.     }
29.     void setMonth(int month){           //
   set method for month
30.         this.month = month;
31.     }
32.     void setYear(int year){           //set method for year
33.         this.year = year;
34.     }
```

```
35.     void display() {           //
method for displaying the date
36.         System.out.print(this.day + "/" + this.month + "/"
" + this.year);
37.     }
38.
39.     public boolean check(int Day, int Month, int Year)
{           //
method to check whether the entered date is correct or no
t
40.         //checking year
41.         int flagYear=0;
42.         int flagMonth=0;
43.         int flagDay=0;
44.         if(Year>2020 || Year <0){
45.             System.out.println("Entered year is incorrect
: (");
46.         }
47.         else{
48.             System.out.println("YEAR : CORRECT");
49.             flagYear=1;
50.         }
51.         //checking month
52.         if(Month>12 || Month<1){
53.             System.out.println("Entered month is incorrec
t : (");
54.         }
55.         else{
56.             System.out.println("MONTH : CORRECT");
57.             flagMonth=1;
58.         }
59.         //checking day
60.         if(Month!=2){
61.             if(Month == 1 || Month == 3 || Month == 5 ||
Month == 7 || Month == 8 || Month == 10 || Month == 12)
{
62.                 if(Day>0 && Day<32){
63.                     System.out.println("DAY : CORRECT");
64.                     flagDay=1;
65.                 }
66.                 else
67.                     System.out.println("Entered day is incorr
ect : (");
68.             }
69.             else{
70.                 if(Day>0 && Day<31){
71.                     System.out.println("DAY : CORRECT");
```

```

72.         flagDay=1;
73.     }
74.     else
75.         System.out.println("Entered day is incorr
ect :");
76.     }
77. }
78. else{
79.     if(Year%4==0 && Year%100!=0){
80.         if(Day>0 && Day<30){
81.             System.out.println("DAY : CORRECT");

82.         flagDay=1;
83.     }
84.     else
85.         System.out.println("Entered day is incorr
ect :");
86.     }
87.     else{
88.         if(Day>0 && Day<29){
89.             System.out.println("DAY : CORRECT");

90.         flagDay=1;
91.     }
92.     else
93.         System.out.println("Entered day is incorr
ect :");
94.     }
95. }
96. if(flagDay==1 && flagMonth==1 && flagYear==1)
97.     return true;        //
returns true if date is valid
98. else
99.     return false;        //
returs false if date is invalid
100. }
101.}
102.class Cricketer{        //
class which represent the Cricketer (players of match)
103.    String playerFirstName;    //player's name
104.    int totalScore,totalWicket;    //player's total-
Score if he/she is batsman OR player's total-
Wicket if he/she is bowler(sum of - score/
wickets from all matches).
105.    boolean rollInTeam;        //
if player is batsman then rollInTeam will be true otherwi
se false.
106.    Cricketer(String playerFirstName, boolean rollInTeam)
{        //construcotr with argument

```

```
107.         this.playerFirstName = playerFirstName;
108.         this.rollInTeam = rollInTeam;
109.         this.totalScore = 0;
110.         this.totalWicket = 0;
111.     }
112.     void setTotalScore(int totalScore){        //
        setter method for totalScore
113.         this.totalScore += totalScore;
114.     }
115.     void setTotalWicket(int totalWicket) {        //
        setter method for totalWicket
116.         this.totalWicket += totalWicket;
117.     }
118.     int getTotalScore() {        //
        getter method for totalScore
119.         return totalScore;
120.     }
121.     int getTotalWicket() {        //
        getter method for totalWicket
122.         return totalWicket;
123.     }
124. }
125. class Match{        //class which represents each Match
126.     Date matchDate = new Date();        //
        represents match Date
127.     int totalScore, totalWickets;        //
        represents (total run scored)/
        (total wickets taken) in this particular match
128.     int[] matchInfo = new int[11];        //
        array for storing player's run/wickets
129.     Cricketer[] players;        //
        array of cricketer object.
130.
131.     Match(Date matchDate, Cricketer[] players){        //
        constructor with arguments (of Objects)
132.         this.totalScore = 0;
133.         this.totalWickets = 0;
134.         this.matchDate = matchDate;
135.         this.players = players;
136.     }
137.
138.     void addPlayerDetails(int index, int scoreOrWicket) {
        //method for adding players into the match
139.         //rollInTeam is batsman ==> rollInTeam = 1 ||
        rollInTeam is bowler ==> rollInTeam = 0
140.         //in array, rollInTeam is batsman ==> 1 ||
        rollInTeam is bowler ==> 0
```

```
141.          //
           in array, first index is rollInTeam, second index is scor
           e/wickets
142.          if(players[index].rollInTeam == true) {
143.              this.players[index].setTotalScore(scoreOrWick
           et);
144.              this.totalScore += scoreOrWicket;
145.          }
146.          else {
147.              this.players[index].setTotalWicket(scoreOrWic
           ket);
148.              this.totalWickets += scoreOrWicket;
149.          }
150.
151.          this.matchInfo[index] = scoreOrWicket;
152.
153.      }
154.
155.      void displayPlayerDetails() {          //
           method for displaying the player's details like runs/
           wickets,roll-In-Team in this match.
156.          for(int i=0;i<11;i++) {
157.              System.out.println("\nPlayer no. : " + (i+1))
           ;
158.              System.out.println("Player Name : " + players
           [i].playerFirstName);
159.              System.out.println("Roll In Team : " + ((play
           ers[i].rollInTeam==false)?"Bowler":"Batsman"));
160.              System.out.println(((players[i].rollInTeam==f
           alse)? ("Wickets taken : ") : ("Score : ")) + matchInfo[i
           ]);
161.          }
162.      }
163.      void displayMatchInfo() {          //
           method for displaying the match-date, total-
           score and total-Wickets
164.          System.out.println("Match date : " + this.matchDa
           te.day + " / " + this.matchDate.month + " / " + this.matc
           hDate.year);
165.          System.out.println("Total Score : " + this.totalsC
           ore);
166.          System.out.println("Total Wickets : " + this.tota
           lWickets);
167.      }
168.  }
169.
170.
171.  public class Main {          //Main class
172.      Scanner sc = new Scanner(System.in);
```

```
173.    Cricketer[] players = new Cricketer[11];           //
      Array of Object Cricketer(players)
174.    ArrayList<Match> matches = new ArrayList<Match>();
      //ArrayList for storing the matches
175.    int day,month,year;
176.
177.    Match addMatch() {           //
      method for adding new match to the ArrayList-matches
178.        System.out.println("Adding details for match number : " + (matches.size()+1));
179.        int scoreOrWicket;       //
      if player is batsman then it will store his/her runs/
      score and if player is bowler then it will store wickets
      taken by him/her
180.        System.out.println("Enter match date : ");
181.
182.        Date d = new Date(day,month,year);
183.        while(true) {
184.            try {
185.                System.out.print("Day : ");
186.                day = sc.nextInt();
187.                System.out.print("Month : ");
188.                month = sc.nextInt();
189.                System.out.print("Year : ");
190.                year = sc.nextInt();
191.                d.setDay(day);
192.                d.setMonth(month);
193.                d.setYear(year);
194.                if(d.check(day,month,year)==false)
195.                    throw new WrongDate("WrongDate exception
      occurred.\n");           //
      throw an object of user defined exception WrongDate
196.            else
197.                break;
198.        } catch (WrongDate ex) {
199.            System.out.println(ex.getMessage());
      //prints the message from WrongDate exception
200.            System.out.println("Enter date again : ")
      ;
201.        }
202.    }
203.
204.    //Adding player details one by one :
205.    Match m = new Match(d,players);
206.    int sumOfWickets = 0;
207.    System.out.println("Enter player details : ");
208.    for(int i=0;i<11;i++) {
209.        System.out.println("For player " + players[i]
      .playerFirstName);
```

```

210.         if(players[i].rollInTeam == true) {
211.             System.out.print("Enter score : ");
212.             scoreOrWicket = sc.nextInt();
213.         }
214.         else {
215.
216.             while (true) {
217.                 try {
218.                     System.out.print("Enter wickets :
219. ");
220.                     scoreOrWicket = sc.nextInt();
221.                     if(scoreOrWicket<=10 && scoreOrWi
222. cket>=0 && (sumOfWickets+scoreOrWicket<=10))
223.                         break;
224.                     else
225.                         throw new InvalidWicket("Total Wi
226. ckets in a match should be in range from 0 to 10 :
227. (\n");           //throw an object of userdefined exception
228.                     } catch (InvalidWicket e) {
229.                         System.out.println("Exception occ
230. urred : " + e.getMessage());           //
231.                         prints then message from InvalidWicket object
232.                         System.out.println("Enter wickets
233. again : ");
234.                     }
235.                 }
236.                 sumOfWickets+=scoreOrWicket;
237.             }
238.             m.addPlayerDetails(i, scoreOrWicket);
239.         }
240.         return m;           //
241.         //returns the recently created match object.
242.     }
243.
244.     void displayMatchInfo() {           //
245.         //method for displaying the match's info which includes mat
246.         //ch-date, total runs and total wickets.
247.         if(matches.size()==0) {           //
248.             //if ArrayList is empty
249.             System.out.println("Record is empty :(\n");
250.             return;
251.         }
252.
253.         System.out.println("Enter match number : " + "(fr
254. om 1 to " + matches.size() + ")");
255.         int index;
256.         Match m1;
257.         do {

```



```

246.         //exception handling : Exception class used -
        ArrayIndexOutOfBoundsException
247.         try {           //try block
248.             index = sc.nextInt();
249.             //
            when user enters invalid match index(i.e less than zero or
            greater than total matches) then exception will be occurred.
250.             if(index<1 || index>matches.size())
251.                 throw new ArrayIndexOutOfBoundsException(
                "Exception occurred : Array index is out of Bounds :
                (\n");
252.             m1 = matches.get(index-1);
253.             break;
254.         } catch (ArrayIndexOutOfBoundsException e1) {
            //catch block
255.             System.out.print(e1.getMessage());
256.             System.out.print("Please enter valid match
            h number : ");
257.
258.         }
259.
260.     } while(true);           //do-
        while loop till condition fulfilled.
261.     m1.displayMatchInfo();           //
        actual method for showing match-information
262. }
263.
264.     void displayMatchPlayerInfo() {           //
        method for displaying the detail of user-
        entered players's name according to the number. which also
        includes his/her scored-runs/wickets-
        taken in that specific match.
265.         if(matches.size()==0) {           //
            if ArrayList is empty
266.             System.out.println("Record is empty :(\n");
267.             return;
268.         }
269.
270.
271.         System.out.println("Enter match number : " + "(from 1 to "
            + matches.size() + ")");
272.         int index;
273.         Match m1;
274.         do {
275.
276.             try {
277.                 index = sc.nextInt();
278.

```

```

279.             if(index<1 || index>matches.size())
280.                 throw new ArrayIndexOutOfBoundsException(
"Exception occurred : Array index is out of Bounds :
(\n");
281.             m1 = matches.get(index-1);
282.             break;
283.         } catch (ArrayIndexOutOfBoundsException e1) {

284.             System.out.print(e1.getMessage());
285.             System.out.print("Please enter valid matc
h number : ");
286.
287.         }
288.
289.     }while(true);
290.     m1.displayPlayerDetails();    //
actual display method
291. }
292.
293.     void displayPlayerInfo() {    //
method for displaying the player's information. which inc
ludes details like which matches he played till now and i
ts corrospoing date and his/her scored-runs/wickets-
taken
294.         if(matches.size()==0){    //
if ArrayList is empty
295.             System.out.println("Not enough records :
(\n");
296.             return;
297.         }
298.
299.         System.out.println("Enter player number : (from 1
to 11)");
300.         int index;
301.
302.         while (true) {
303.             try {
304.                 index = sc.nextInt();
305.                 //
Exception handling : when user enters the player number o
ther then from 1 to 11 -
Exception name : ArrayIndexOutOfBoundsException
306.                 if(index<1 || index>11)
307.                     throw new ArrayIndexOutOfBoundsException(
"Exception occurred : Array index is out of Bounds :
(\n");
308.                 Match m1;

```

```
309.                System.out.println("\t" + players[index-1
    ].playerFirstName + "\t" + ((players[index-1].rollInTeam=
    =false)? "Bowler" : "Batsman"));
310.                for(int i=0;i<matches.size();i++) {
311.                    m1 = matches.get(i);
312.                    System.out.print("\n[" + (i+1) + "]" "
    ");
313.                    m1.matchDate.display();
314.                    if(players[index-1].rollInTeam == fal
    se) {
315.                        System.out.println(" Wickets : "
    + m1.matchInfo[index-1]);
316.                    }
317.                    else {
318.                        System.out.println(" Score : " +
    m1.matchInfo[index-1]);
319.                    }
320.                }
321.                break;
322.            } catch (ArrayIndexOutOfBoundsException e1) {
323.                System.out.print(e1.getMessage());
324.                System.out.print("Please enter valid play
    er number : ");
325.            }
326.        }
327.    }
328. }
329.
330. void displayPlayerTopInfo() { //
    Method for displaying only top matches of the given playe
    r on the basis of his/
    her performance. this details includes match-
    date and player's scored-runs/wickets-taken
331.    if(matches.size())<4) { //
    if ArrayList has not more then 3 entries then there is no
    significance of showing the top 3 player information. be
    cause when there are atleast 4 matches then only we can s
    how the top 3 matches of that player on the basis of play
    er's performace in each match.
332.        System.out.println("Not enough records :
    (\n");
333.        return;
334.    }
335.    System.out.println("Enter player number : (from 1
    to 11)");
336.    int index;
337.
338.    while (true) {
```

```
339.         try {
340.             index = sc.nextInt();
341.
342.             if(index<1 || index>11)
343.                 throw new ArrayIndexOutOfBoundsException(
"Exception occurred : Array index is out of Bounds :
(\n");
344.
345.             break;
346.         } catch (ArrayIndexOutOfBoundsException e1) {
347.             System.out.print(e1.getMessage());
348.             System.out.print("Please enter valid play
er number : ");
349.
350.         }
351.     }
352.
353.     Match m1;
354.
355.     String[]
[] top3Matches = new String[matches.size()][2];
356.
357.     //collecting data
358.     for(int i=0;i<matches.size();i++) {
359.         m1 = matches.get(i);
360.         top3Matches[i]
[0] = m1.matchDate.day + "/" + m1.matchDate.month + "/" +
m1.matchDate.year;
361.         top3Matches[i]
[1] = "" + m1.matchInfo[index-1];
362.     }
363.
364.     //sorting matches
365.     for(int pick = 0;pick<matches.size();pick++) {
366.         for(int comp = pick+1;comp<matches.size();com
p++) {
367.             if(Integer.parseInt(top3Matches[pick]
[1])<Integer.parseInt(top3Matches[comp][1])) {
368.                 String temp1 = top3Matches[pick]
[1];
369.                 top3Matches[pick]
[1] =top3Matches[comp][1];
370.                 top3Matches[comp][1] = temp1;
371.                 temp1 = top3Matches[pick][0];
372.                 top3Matches[pick]
[0] =top3Matches[comp][0];
373.                 top3Matches[comp][0] = temp1;
374.             }
```

```

375.         }
376.     }
377.
378.         //display top matches of that specific player
379.         System.out.println("Top matches of " + players[index-1].playerFirstName + " : \nDate " + ((players[index-1].rollInTeam==true)? " Score ":" Wickets "));
380.         for(int i=0;i<(3<matches.size()? 3:matches.size());i++) {
381.             System.out.println(top3Matches[i][0] + " " + top3Matches[i][1]);
382.         }
383.     }
384.
385.     void topPlayers() { //
        method for displaying the top 3 batsmans and top 3 bowler
        s from all match's data. batsman are rated on the basis o
        f total score in all matches played and bowlers are rated
        on the basis of total wickets taken in all matches playe
        d.
386.         if(matches.size()==0){ //
            if ArrayList is empty
387.             System.out.println("Not enough records :
(\n");
388.             return;
389.         }
390.
391.         int[][] topBatsman = new int[11][2]; //
            2d integer array
392.         int[][] topBowlers = new int[11][2]; //
            2d integer array
393.         int count1=0,count2=0;
394.
395.         //
            collecting the data of all players and storing it in the
            about arrays.
396.         for(int i=0;i<11;i++) {
397.             if(players[i].rollInTeam == true) {
398.                 topBatsman[count1][0] = i;
399.                 topBatsman[count1]
                    [1] = players[i].totalScore;
400.                 count1++;
401.             }
402.             else {
403.                 topBowlers[count2][0] = i;
404.                 topBowlers[count2]
                    [1] = players[i].totalWicket;
405.                 count2++;
406.             }

```

```
407.         }
408.
409.         //sorting of batsmen
410.         for(int pick = 0;pick<count1;pick++) {
411.             for(int comp = pick+1;comp<count1;comp++) {
412.                 if(topBatsman[pick][1] < topBatsman[comp]
[1]) {
413.                     int temp = topBatsman[pick][1];
414.                     topBatsman[pick]
[1] = topBatsman[comp][1];
415.                     topBatsman[comp][1] = temp;
416.                     temp = topBatsman[pick][0];
417.                     topBatsman[pick]
[0] = topBatsman[comp][0];
418.                     topBatsman[comp][0] = temp;
419.                 }
420.             }
421.         }
422.
423.         //display batsmen
424.         System.out.println("Batsman : \nName      Score");
425.
426.         for(int i=0;i<3;i++) {
427.             if(i<count1)
428.                 System.out.println(players[topBatsman[i]
[0]].playerFirstName + "      " + topBatsman[i][1]);
429.
430.
431.         //sorting of bowler
432.         for(int pick = 0;pick<count2;pick++) {
433.             for(int comp = pick+1;comp<count2;comp++) {
434.                 if(topBowlers[pick][1] < topBowlers[comp]
[1]) {
435.                     int temp = topBowlers[pick][1];
436.                     topBowlers[pick]
[1] = topBowlers[comp][1];
437.                     topBowlers[comp][1] = temp;
438.                     temp = topBowlers[pick][0];
439.                     topBowlers[pick]
[0] = topBowlers[comp][0];
440.                     topBowlers[comp][0] = temp;
441.                 }
442.             }
443.         }
444.
445.         //display bowlers
446.         System.out.println("Bowlers : \nName      Wickets")
;
```

```
447.         for(int i=0;i<3;i++) {
448.             if(i<count2)
449.                 System.out.println(players[topBowlers[i]
450. [0]].playerFirstName + "      " + topBowlers[i][1]);
451.         }
452.     }
453.
454.     public static void main(String[] args) {          //
main method
455.         Main project = new Main();
456.         String name;
457.         boolean rollInTeam;
458.
459.         System.out.println("Enter player details : \n");

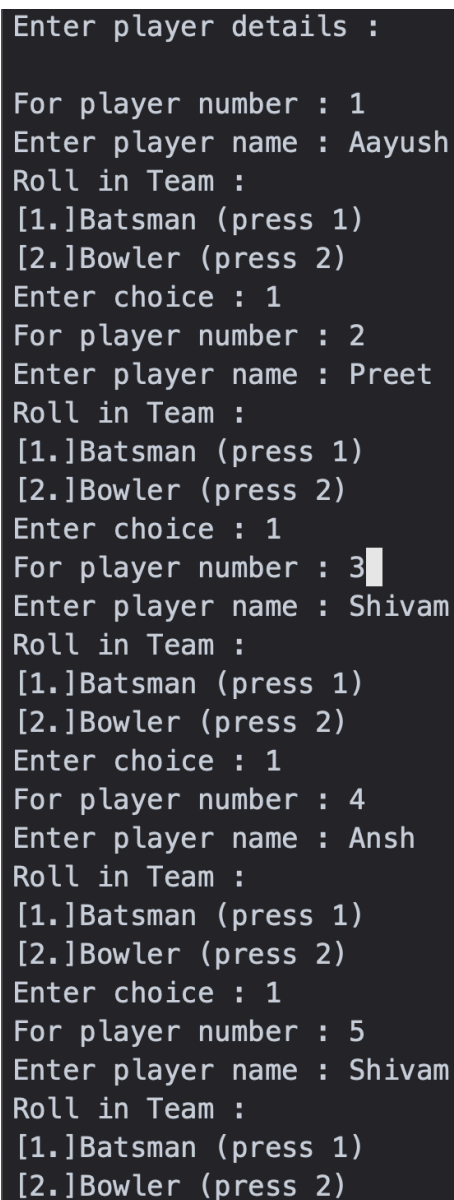
460.         //userdefined players :
461.         for(int i=0;i<11;i++) {
462.             //user Input
463.             System.out.println("For player number : " + (
i+1));
464.             System.out.print("Enter player name : ");
465.             name = project.sc.next();
466.             System.out.print("Roll in Team : \n[1.]Batsma
n (press 1)\n[2.]Bowler (press 2)\nEnter choice : ");
467.             int choice;
468.             do {
469.                 choice = project.sc.nextInt();
470.                 if(choice!=1 && choice!=2)
471.                     System.out.print("Invalid Input :
(\nEnter choice : ");
472.             }while(choice!=1 && choice!=2);
473.             rollInTeam = (choice==1)?true:false;      //
assignes true if player is batsman otherwise false.
474.             project.players[i] = new Cricketer(name, roll
InTeam);
475.         }
476.
477.         /*
478.         //pre defined sample players :
479.         project.players[0] = new Cricketer("a", true);
480.         project.players[1] = new Cricketer("b", true);
481.         project.players[2] = new Cricketer("e", true);
482.         project.players[3] = new Cricketer("d", true);
483.         project.players[4] = new Cricketer("f", true);
484.         project.players[5] = new Cricketer("g", true);
485.         project.players[6] = new Cricketer("h", false);
486.         project.players[7] = new Cricketer("i", false);
487.         project.players[8] = new Cricketer("j", false);
```

```
488.         project.players[9] = new Cricketer("k", false);
489.         project.players[10] = new Cricketer("l", false);
490.     */
491.
492.     int choice = 0;
493.     while(choice!=7) {           //MenuDriven
494.         System.out.println("Menu : ");
495.         System.out.println("\n[1.] Add match");
496.         System.out.println("[2.] Display match information");
497.         System.out.println("[3.] Display match player s information");
498.         System.out.println("[4.] Display player information (all match)");
499.         System.out.println("[5.] Display player information (top 3 match)");
500.         System.out.println("[6.] Display top 3 batsmen and bowlers");
501.         System.out.println("[7.] Exit");
502.         System.out.print("Enter Choice : ");
503.         choice = project.sc.nextInt();
504.         switch(choice) {
505.             case 1 :
506.                 project.matches.add(project.addMatch(
507.                 ));
508.                 break;
509.             case 2 :
510.                 project.displayMatchInfo();
511.                 break;
512.             case 3 :
513.                 project.displayMatchPlayerInfo();
514.                 break;
515.             case 4 :
516.                 project.displayPlayerInfo();
517.                 break;
518.             case 5 :
519.                 project.displayPlayerTopInfo();
520.                 break;
521.             case 6 :
522.                 project.topPlayers();
523.                 break;
524.             case 7 :
525.                 System.out.println("THANK YOU");
526.                 break;
527.             case 7 :
528.                 System.out.println("THANK YOU");
529.                 break;
530.         }
```



```
531.                                     break;
532.
533.                                     default :
534.                                     System.out.println("Invalid Choice :
    ("");
535.                                     break;
536.                                     }
537.                                     }
538.
539.                                     }
540.
541. }
```

Screen Shots :



```
Enter player details :

For player number : 1
Enter player name : Aayush
Roll in Team :
[1.]Batsman (press 1)
[2.]Bowler (press 2)
Enter choice : 1
For player number : 2
Enter player name : Preet
Roll in Team :
[1.]Batsman (press 1)
[2.]Bowler (press 2)
Enter choice : 1
For player number : 3
Enter player name : Shivam
Roll in Team :
[1.]Batsman (press 1)
[2.]Bowler (press 2)
Enter choice : 1
For player number : 4
Enter player name : Ansh
Roll in Team :
[1.]Batsman (press 1)
[2.]Bowler (press 2)
Enter choice : 1
For player number : 5
Enter player name : Shivam
Roll in Team :
[1.]Batsman (press 1)
[2.]Bowler (press 2)
```

```
Enter choice : 1
For player number : 6
Enter player name : Harshal
Roll in Team :
[1.]Batsman (press 1)
[2.]Bowler (press 2)
Enter choice : 1
For player number : 7
Enter player name : Yash
Roll in Team :
[1.]Batsman (press 1)
[2.]Bowler (press 2)
Enter choice : 2
For player number : 8
Enter player name : Smit
Roll in Team :
[1.]Batsman (press 1)
[2.]Bowler (press 2)
Enter choice : 2
For player number : 9
Enter player name : Kedar
Roll in Team :
[1.]Batsman (press 1)
[2.]Bowler (press 2)
Enter choice : 2
For player number : 10
Enter player name : Jay
Roll in Team :
[1.]Batsman (press 1)
[2.]Bowler (press 2)
Enter choice : 2
```

```
For player number : 11
Enter player name : Karnik
Roll in Team :
[1.]Batsman (press 1)
[2.]Bowler (press 2)
Enter choice : 2
Menu :

[1.] Add match
[2.] Display match information
[3.] Display match players information
[4.] Display player information (all match)
[5.] Display player information (top 3 match)
[6.] Display top 3 batsmen and bowlers
[7.] Exit
Enter Choice : 1
Adding details for match number : 1
Enter match date :
Day : 12
Month : 10
Year : 2019
YEAR : CORRECT
MONTH : CORRECT
DAY : CORRECT
Enter player details :
For player Aayush
Enter score : 10
For player Preet
Enter score : 20
For player Shivam
Enter score : 30
```

```
For player Ansh
Enter score : 40
For player Shivam
Enter score : 50
For player Harshal
Enter score : 60
For player Yash
Enter wickets : 1
For player Smit
Enter wickets : 100
Exception occurred : Total Wickets in a match should be in range
    from 0 to 10 :(

Enter wickets again :
Enter wickets : 3
For player Kedar
Enter wickets : 0
For player Jay
Enter wickets : 2
For player Karnik
Enter wickets : 1
Menu :

[1.] Add match
[2.] Display match information
[3.] Display match players information
[4.] Display player information (all match)
[5.] Display player information (top 3 match)
[6.] Display top 3 batsmen and bowlers
[7.] Exit
Enter Choice : 1
```

```
Adding details for match number : 2
Enter match date :
Day : 100
Month : 13
Year : 3000
Entered year is incorrect :(
Entered month is incorrect :(
Entered day is incorrect :(
WrongDate exception occurred.
█
Enter date again :
Day : 12
Month : 12
Year : 2019
YEAR : CORRECT
MONTH : CORRECT
DAY : CORRECT
Enter player details :
For player Aayush
Enter score : 3
For player Preet
Enter score : 4
For player Shivam
Enter score : 5
For player Ansh
Enter score : 6
For player Shivam
Enter score : 7
For player Harshal
Enter score : 8
For player Yash
```

```
Enter wickets : 5
For player Smit
Enter wickets : 5
For player Kedar
Enter wickets : 0
For player Jay
Enter wickets : 0
For player Karnik
Enter wickets : 12
Exception occurred : Total Wickets in a match should be in range
    from 0 to 10 :(

Enter wickets again :
Enter wickets : 0
Menu :

[1.] Add match
[2.] Display match information
[3.] Display match players information
[4.] Display player information (all match)
[5.] Display player information (top 3 match)
[6.] Display top 3 batsmen and bowlers
[7.] Exit
Enter Choice : 1
Adding details for match number : 3
Enter match date :
Day : 1
Month : 1
Year : 2020
YEAR : CORRECT
MONTH : CORRECT
```

```
DAY : CORRECT
Enter player details :
For player Aayush
Enter score : 0
For player Preet
Enter score : 10
For player Shivam
Enter score : 22
For player Ansh
Enter score : 100
For player Shivam
Enter score : 1
For player Harshal
Enter score : 50
For player Yash
Enter wickets : 0
For player Smit
Enter wickets : 1
For player Kedar
Enter wickets : 1
For player Jay
Enter wickets : 3
For player Karnik
Enter wickets : 2
Menu :

[1.] Add match
[2.] Display match information
[3.] Display match players information
[4.] Display player information (all match)
[5.] Display player information (top 3 match)
```

```
[6.] Display top 3 batsmen and bowlers
[7.] Exit
Enter Choice : 1
Adding details for match number : 4
Enter match date :
Day : 20
Month : 6
Year : 2020
YEAR : CORRECT
MONTH : CORRECT
DAY : CORRECT
Enter player details :
For player Aayush
Enter score : 100
For player Preet
Enter score : 20
For player Shivam
Enter score : 10
For player Ansh
Enter score : 5
For player Shivam
Enter score : 0
For player Harshal
Enter score : 1
For player Yash
Enter wickets : 5
For player Smit
Enter wickets : 1
For player Kedar
Enter wickets : 1
For player Jay
```



```
Enter wickets : 1
For player Karnik
Enter wickets : 0
Menu :

[1.] Add match
[2.] Display match information
[3.] Display match players information
[4.] Display player information (all match)
[5.] Display player information (top 3 match)
[6.] Display top 3 batsmen and bowlers
[7.] Exit
Enter Choice : 2
Enter match number : (from 1 to 4)
3
Match date : 1 / 1 / 2020
Total Score : 183
Total Wickets : 7
Menu :

[1.] Add match
[2.] Display match information
[3.] Display match players information
[4.] Display player information (all match)
[5.] Display player information (top 3 match)
[6.] Display top 3 batsmen and bowlers
[7.] Exit
Enter Choice : 3
Enter match number : (from 1 to 4)
2
```

Player no. : 1
Player Name : Aayush
Roll In Team : Batsman
Score : 3

Player no. : 2
Player Name : Preet
Roll In Team : Batsman
Score : 4

Player no. : 3
Player Name : Shivam
Roll In Team : Batsman
Score : 5

Player no. : 4
Player Name : Ansh
Roll In Team : Batsman
Score : 6

Player no. : 5
Player Name : Shivam
Roll In Team : Batsman
Score : 7

Player no. : 6
Player Name : Harshal
Roll In Team : Batsman
Score : 8

Player no. : 7

```
Player no. : 7
Player Name : Yash
Roll In Team : Bowler
Wickets taken : 5

Player no. : 8
Player Name : Smit
Roll In Team : Bowler
Wickets taken : 5

Player no. : 9
Player Name : Kedar
Roll In Team : Bowler
Wickets taken : 0

Player no. : 10
Player Name : Jay
Roll In Team : Bowler
Wickets taken : 0

Player no. : 11
Player Name : Karnik
Roll In Team : Bowler
Wickets taken : 0
Menu :

[1.] Add match
[2.] Display match information
[3.] Display match players information
[4.] Display player information (all match)
```

```
[5.] Display player information (top 3 match)
[6.] Display top 3 batsmen and bowlers
[7.] Exit
Enter Choice : 4
Enter player number : (from 1 to 11)
3
    Shivam  Batsman

[1] 12/10/2019 Score : 30
[2] 12/12/2019 Score : 5
[3] 1/1/2020 Score : 22
[4] 20/6/2020 Score : 10
Menu :

[1.] Add match
[2.] Display match information
[3.] Display match players information
[4.] Display player information (all match)
[5.] Display player information (top 3 match)
[6.] Display top 3 batsmen and bowlers
[7.] Exit
Enter Choice : 4
Enter player number : (from 1 to 11)
9
    Kedar  Bowler

[1] 12/10/2019 Wickets : 0
```

```
[2] 12/12/2019 Wickets : 0

[3] 1/1/2020 Wickets : 1

[4] 20/6/2020 Wickets : 1
Menu :

[1.] Add match
[2.] Display match information
[3.] Display match players information
[4.] Display player information (all match)
[5.] Display player information (top 3 match)
[6.] Display top 3 batsmen and bowlers
[7.] Exit
Enter Choice : 5
Enter player number : (from 1 to 11)
2
Top matches of Preet :
Date          Score
12/10/2019    20
20/6/2020     20
1/1/2020      10
Menu :

[1.] Add match
[2.] Display match information
[3.] Display match players information
[4.] Display player information (all match)
[5.] Display player information (top 3 match)
[6.] Display top 3 batsmen and bowlers
[7.] Exit
```

```
[7.] Exit
Enter Choice : 5
Enter player number : (from 1 to 11)
11
Top matches of Karnik :
Date          Wickets
1/1/2020      2
12/10/2019    1
12/12/2019    0
Menu :

[1.] Add match
[2.] Display match information
[3.] Display match players information
[4.] Display player information (all match)
[5.] Display player information (top 3 match)
[6.] Display top 3 batsmen and bowlers
[7.] Exit
Enter Choice : 6
Batsman :
Name    Score
Ansh     151
Harshal  119
Aayush   113
Bowlers :
Name    Wickets
Yash     11
Smit     10
Jay       6
Menu :

[1.] Add match
[2.] Display match information
[3.] Display match players information
[4.] Display player information (all match)
[5.] Display player information (top 3 match)
[6.] Display top 3 batsmen and bowlers
[7.] Exit
Enter Choice : 7
THANK YOU
```

✓ Run Succeeded | Time 576 ms | Peak Memory 36.6M

Symbol ↕ | Tabs: 4 ↕ | Line 1, Column