

Solve the six problems by using Java Programming.

100 Marks.

1 – 10 marks

2 – 6 – 30 marks

3 – 4 - 35 marks

5- 25 marks

1.

Requirement 1:

Let's start off by creating two Player objects and check whether they are equal.

1. Create a Player Class with the following attributes:

Member Field Name	Type
name	String
dateOfBirth	java.util.Date
skill	String
numberOfMatches	Integer
runs	Integer
wickets	Integer
nationality	String
powerRating	Double

2. Mark all the attributes as private

3. Create / Generate appropriate Getters & Setters

4. Add a default constructor and a parameterized constructor to take in all attributes in the given order:

Player(String name , Date dateOfBirth , String skill , Integer

numberOfMatches , Integer runs , Integer wickets , String nationality , Double powerRating)

5. When the “player” object is printed, it should display the following details:
[Override the toString method]

Print format:

Name:"name"

Date of Birth:"dateOfBirth"

Skill:"skill"

Number of Matches:"numberOfMatches"

Runs:"runs"

Wickets:"wickets"

Nationality:"nationality"

PowerRating:"powerRating"

6. Two players are considered same if they have the same name, skill, and nationality. Implement the logic in the appropriate function. (Case – Insensitive) **[Override the equals method]**

The input format consists of mail details separated by comma in the below order,

(name,dateOfBirth,skill,numberOfMatches,runs,wickets,nationality,powerRating)

The Input to your program would be details of two players, you need to display their details as given in "5th point(refer above)" and compare the two players and display if the Players are same or different.

Note: There is an empty line between display statements. Print the empty lines in the main function.

Display one digit after the decimal point for Double datatype.

Sample INPUT & OUTPUT 1:

Enter player 1 detail:

MSD,07-07-1981,WK&BAT,300,10000,2,Indian,4.9

Enter player 2 detail:

MSD,07-07-1981,WK&BAT,300,10000,2,Indian,4.9

Player 1:

Name:MSD
Date of Birth:07-07-1981
Skill:WK&BAT
Number of Matches:300
Runs:10000
Wickets:2
Nationality:Indian
Power Rating:4.9

Player 2:
Name:MSD
Date of Birth:07-07-1981
Skill:WK&BAT
Number of Matches:300
Runs:10000
Wickets:2
Nationality:Indian
Power Rating:4.9

Player 1 is same as Player 2

Sample INPUT & OUTPUT 2:

Enter player 1 detail:
MSD,07-07-1981,WK&BAT,300,10000,2,Indian,4.9
Enter player 2 detail:
ABD,17-02-1984,WK&BAT,300,10000,3,South African,4.9

Player 1:
Name:MSD
Date of Birth:07-07-1981
Skill:WK&BAT
Number of Matches:300
Runs:10000
Wickets:2
Nationality:Indian
Power Rating:4.9

Player 2:
Name:ABD

Date of Birth:17-02-1984

Skill:WK&BAT

Number of Matches:300

Runs:10000

Wickets:3

Nationality:South African

Power Rating:4.9

Player 1 and Player 2 are different

2.

In this requirement, creating a team and add players to it. Start with creating a team and use menu-driven approach to add, remove, display details of the player in the team.

a) Create a Class **Player** with the following attributes:

Member Field Name	Type
name	String
dateOfBirth	java.util.Date / java.time.LocalDate
skill	String
numberOfMatches	Integer
runs	Integer
wickets	Integer
nationality	String
powerRating	Double

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **public Player(String name, Date dateOfBirth, String skill, Integer numberOfMatches, Integer runs, Integer wickets, String nationality, Double powerRating).**

b) Create a Class **Team** with the following attributes:

Member Field Name	Type
name	String
playerList	List<Player>

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **public Team(String name, List<Player> playerList).** In constructor pass the playerList value as an empty list. Only one team will be present at a time.

c) Create the following static method in Player class,

Method Name	Description
public static Player createPlayer(String detail)	This method accepts a String which contains player details separated by commas. Split the details and create a player object from the details and return it.

The player details should be given as a comma-separated value in the below order, **name,dateOfBirth,skill,numberOfMatches,runs,wickets,nationality,powerRating**

d) Create the following methods in Team class,

Method Name	Description
public void addPlayerToTeam(Player player)	This method accepts a Player object and add the player to the player list of the current team.
public Boolean removePlayerFromTeam(String name)	This method will get the name of the player and delete the player with the specified name from the current team. If a player with the given name is found, delete the player and return true . If a player with the name is not found return false . The players name are unique.
public void displayPlayers()	This method will display the player list in the current team. If the player list is empty display " No players to show ". If not empty, display "Players in [team name]" and display all the player details in the specified format. Where [team name] specifies the name of the team.

After deletion, if true is returned print "**Player successfully deleted**", else print "**Player not found in the team**". After adding a player to the team, print "**Player successfully added**".

Note: The above print statements should be present in the main method.

When the "player" object is printed, it should display the following format
Print format:

System.out.printf("%-15s%-15s%-10s%-15s%-10s%-10s%-15s%-10s\n", "Name","Date of birth","Skill","No of matches","Runs","Wickets","Nationality","Rating"). Display 1 digit after decimal point in Double.

Sample Input and Output:

Enter the name of the Team:

Royal Challengers Bangalore

- 1.Add Player
- 2.Delete Player
- 3.Display Players
- 4.Exit

Enter your choice:

3

No players to show

- 1.Add Player
- 2.Delete Player
- 3.Display Players
- 4.Exit

Enter your choice:

1

Enter the details of player in CSV format:

Virat Kohli,05-11-1988,Batsman,149,4418,4,India,4.7

Player successfully added

- 1.Add Player
- 2.Delete Player
- 3.Display Players
- 4.Exit

Enter your choice:

1

Enter the details of player in CSV format:

Ab de Villiers,7-02-1984,Batsman,129,3473,0,S Africa,4.7

Player successfully added

- 1.Add Player
- 2.Delete Player
- 3.Display Players
- 4.Exit

Enter your choice:

1

Enter the details of player in CSV format:

Bhuvneshwar,05-02-1990,Bowler,90,158,111,India,4.1

Player successfully added

- 1.Add Player

- 2.Delete Player
- 3.Display Players
- 4.Exit

Enter your choice:

1

Enter the details of player in CSV format:

Mitchell Stark,30-01-1990,Bowler,27,96,34,Australia,4.1

Player successfully added

- 1.Add Player
- 2.Delete Player
- 3.Display Players
- 4.Exit

Enter your choice:

2

Enter the name of the player to be deleted:

MS Dhoni

Player not found in the team

- 1.Add Player
- 2.Delete Player
- 3.Display Players
- 4.Exit

Enter your choice:

2

Enter the name of the player to be deleted:

Bhuvneshwar

Player successfully deleted

- 1.Add Player
- 2.Delete Player
- 3.Display Players
- 4.Exit

Enter your choice:

3

Players in:Royal Challengers Bangalore

Name	Date of birth	Skill	No of matches	Runs	Wickets	Nationality	Rating	
Virat Kohli	05-11-1988	Batsman	149	4418	4	India	4.7	
Ab de Villiers	07-02-1984	Batsman	129	3473	0	Africa	4.7	S
Mitchell Stark	30-01-							

1990 Bowler 27 96 34 Australia 4.1

1.Add Player

2.Delete Player

3.Display Players

4.Exit

Enter your choice:

4

3.

In this requirement develop a feature in which you can search a List of Players by nationality, dateOfBirth or powerRating.

a) Create a Class Player with the following attributes:

Member Field Name	Type
name	String
dateOfBirth	java.util.Date / LocalDate
skill	String
numberOfMatches	Integer
runs	Integer
wickets	Integer
nationality	String
powerRating	Double

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: Player(String name, Date dateOfBirth, String skill, Integer numberOfMatches, Integer runs,Integer wickets,String nationality,Double powerRating)

b) Create a class **PlayerBO** with the following methods,

Method Name	Description
public List<Player> findPlayer(List<Player> playerList,String nationality)	This method accepts a list of players and Nationality as arguments and returns a list of players that match with given Nationality.
public List<Player> findPlayer(List<Player> playerList,Date dateOfBirth)	This method accepts a list of players and date of birth as arguments and returns a list of players who were born on given date.

<pre>public List<Player> findPlayer(List<Player> playerList,Double powerRating)</pre>	This method accepts a list of players and power rating as arguments, then find all the players with the given power rating from the player list and return the list of players with the specified power rating.
---	---

The player details should be given as a comma-separated value in the below order, name,dateOfBirth,skill,numberOfMatches,runs,wickets,nationality,powerRating

Get the number of player and the player details, build a player list and perform the search by nationality,dateOfBirth or powerRating.

When the “player” object is printed, it should display the following details

Print format:

```
System.out.format("%-15s %-15s %-15s %-20s %-15s %-15s %-15s
%s\n","Name","Date of Birth","Skill","Number of
Wickets","Runs","Wickets","Nationality","Power Rating");
```

Note: The player lists are displayed in the main method.

If any other choice is selected, display **"Invalid choice"**

Display one digit after the decimal point for Double Datatype.

Sample Input and Output 1:

Enter the number of Players:

4

MSD,07-07-1981,WK&BAT,300,10000,2,Indian,4.9

ABD,17-02-1984,WK&BAT,300,10000,3,South African,4.9

SRaina,27-11-1986,BAT,100,5000,50,Indian,4.5

Maxwell,14-10-1988,ALLROUNDER,200,4000,50,Australian,4.5

Enter a search type:

1.By Nationality

2.By Date of Birth

3.By Power Rating

1

Enter the Nationality:

Australian

Name	Date of Birth	Skill	Number of Matches	Runs	Wickets	Nationality	Power Rating
------	---------------	-------	-------------------	------	---------	-------------	--------------

Maxwell 14-10-1988 ALLROUNDER 200 4000 50 Australian 4.5

Sample Input and Output 2:

Enter the number of Players:

4

MSD,07-07-1981,WK&BAT,300,10000,2,Indian,4.9

ABD,17-02-1984,WK&BAT,300,10000,3,South African,4.9

SRaina,27-11-1986,BAT,100,5000,50,Indian,4.5

Maxwell,14-10-1988,ALLROUNDER,200,4000,50,Australian,4.5

Enter a search type:

1.By Nationality

2.By Date of Birth

3.By Power Rating

2

Enter the Date of Birth:

07-07-1981

Name	Date of Birth	Skill	Number of Matches	Runs	Wickets	Nationality	Power Rating
MSD	07-07-1981	WK&BAT	300	10000	2	Indian	4.9

Sample Input and Output 3:

Enter the number of Players:

4

MSD,07-07-1981,WK&BAT,300,10000,2,Indian,4.9

ABD,17-02-1984,WK&BAT,300,10000,3,South African,4.9

SRaina,27-11-1986,BAT,100,5000,50,Indian,4.5

Maxwell,14-10-1988,ALLROUNDER,200,4000,50,Australian,4.2

Enter a search type:

1.By Nationality

2.By Date of Birth

3.By Power Rating

3

Enter the Power Rating:

4.5

Name	Date of Birth	Skill	Number of Matches	Runs	Wickets	Nationality	Power Rating
SRaina	27-11-1986	BAT	100	5000	50	Indian	4.5

Sample Input and Output 4:

Enter the number of Players:

4

MSD,07-07-1981,WK&BAT,300,10000,2,Indian,4.9

ABD,17-02-1984,WK&BAT,300,10000,3,South African,4.9

SRaina,27-11-1986,BAT,100,5000,50,Indian,4.5

Maxwell,14-10-1988,ALLROUNDER,200,4000,50,Australian,4.2

Enter a search type:

1.By Nationality

2.By Date of Birth

3.By Power Rating

4

Invalid choice

4.

In this requirement, you need to sort the list of players based on number of matches played, runs scored or powerRating.

a) Create a Class Player with the following attributes:

Member Field Name	Type
name	String
dateOfBirth	java.util.Date / LocalDate
skill	String
numberOfMatches	Integer
runs	Integer
wickets	Integer
nationality	String
powerRating	Double

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **Player(String name, java.util.Date dateOfBirth, String skill, Integer numberOfMatches, Integer runs, Integer wickets, String nationality, Double powerRating)**

b) Create the following static methods in the Player class,

Method Name	Description
static Player createPlayer(String detail)	This method accepts a String and returns a Player object. The player detail separated by commas is passed as value. This method will split the details and creates a player object and returns it.

The Player details should be given as a comma-separated value in the below order, name, dateOfBirth, skill, numberOfMatches, runs, wickets, nationality, powerRating

c) The Player class should implement the **Comparable** interface which sorts the Player list based on the number of matches. While comparing, all the numberOfMatches attributes in the list are unique.

d) Create a class **PowerRatingComparator** which implements Comparator

interface and sort the Player list based on powerRating. While comparing, all the powerRating attributes in the list are unique.

e) Create a class **RunComparator** which implements Comparator interface and sort the Player list based on the runs. While comparing, all the runs attributes in the list are unique.

Note: Step d) & e) can implement the Lambda Expression(Optional).

When the “player” object is printed, it should display the following details

Print format:

```
System.out.format("%-15s %-15s %-15s %-15s %-10s %-10s %-15s %s\n",  
"Name","Date of birth","Skill","No of  
matches","Runs","wickets","Nationality","Power rating");
```

Display one digit after decimal point for Double datatype.

Sample Input and Output 1:

Enter the number of the players:

5

MS Dhoni,07-07-1981,Batsman,159,3561,0,India,4.4

Virat Kohli,05-11-1988,Batsman,149,4418,4,India,4.7

Ab de Villiers,7-02-1984,Batsman,129,3473,0,S Africa,4.6

Mitchell Starc,30-01-1990,Bowler,27,96,34,Australia,4

Bhuvneshwar,05-02-1990,Bowler,90,158,111,India,4.1

Enter a type to sort:

1.Sort by number of matches played

2.Sort by runs scored

3.Sort by power rating

1

Name	Date of birth	Skill	No of matches	Runs	wickets	Nationality	Power rating	
Mitchell Starc	30-01-1990	Bowler	27		96	34	Australia	4.0
Bhuvneshwar	05-02-1990	Bowler	90		158	111	India	4.1
Ab de Villiers	07-02-1984	Batsman	129		3473	0	S Africa	4.6
Virat Kohli	05-11-1988	Batsman	149		4418	4	India	4.7

MS Dhoni	07-07-					
1981	Batsman	159	3561	0	India	4.4

Sample Input and Output 2:

Enter the number of the players:

5

MS Dhoni,07-07-1981,Batsman,159,3561,0,India,4.4
Virat Kohli,05-11-1988,Batsman,149,4418,4,India,4.7
Ab de Villiers,7-02-1984,Batsman,129,3473,0,S Africa,4.6
Mitchell Starc,30-01-1990,Bowler,27,96,34,Australia,4
Bhuvneshwar,05-02-1990,Bowler,90,158,111,India,4.1

Enter a type to sort:

- 1.Sort by number of matches played
- 2.Sort by runs scored
- 3.Sort by power rating

2

Name	Date of birth	Skill	No of matches	Runs	wickets	Nationality	Power rating
Mitchell Starc	30-01-1990	Bowler	27	96	34	Australia	4.0
Bhuvneshwar	05-02-1990	Bowler	90	158	111	India	4.1
Ab de Villiers	07-02-1984	Batsman	129	3473	0	S Africa	4.6
MS Dhoni	07-07-1981	Batsman	159	3561	0	India	4.4
Virat Kohli	05-11-1988	Batsman	149	4418	4	India	4.7

Sample Input and Output 3:

Enter the number of the players:

5

MS Dhoni,07-07-1981,Batsman,159,3561,0,India,4.4
Virat Kohli,05-11-1988,Batsman,149,4418,4,India,4.7
Ab de Villiers,7-02-1984,Batsman,129,3473,0,S Africa,4.6
Mitchell Starc,30-01-1990,Bowler,27,96,34,Australia,4
Bhuvneshwar,05-02-1990,Bowler,90,158,111,India,4.1

Enter a type to sort:

1.Sort by number of matches played

2.Sort by runs scored

3.Sort by power rating

3

Name	Date of birth	Skill	No of			
matches	Runs	wickets	Nationality	Power rating		
Mitchell Starc	30-01-1990	Bowler	27	96	34	Australia 4.0
Bhuvneshwar	05-02-1990	Bowler	90	158	111	India 4.1
MS Dhoni	07-07-1981	Batsman	159	3561	0	India 4.4
Ab de Villiers	07-02-1984	Batsman	129	3473	0	S Africa 4.6
Virat Kohli	05-11-1988	Batsman	149	4418	4	India 4.7

In this requirement, given a list of players you need to find the number of players playing for a country using Map.

a) Create a Class Player with the following attributes:

Member Field Name	Type
name	String
dateOfBirth	java.util.Date / LocalDate
skill	String
numberOfMatches	Integer
runs	Integer
wickets	Integer
nationality	String
powerRating	Double

Mark all the attributes as private, Create / Generate appropriate Getters & Setters. Add a default constructor and a parameterized constructor to take in all attributes in the given order:

Player(String name, java.util.Date dateOfBirth, String skill, Integer numberOfMatches, Integer runs, Integer wickets, String nationality, Double powerRating)

b) Create the following static methods in the **Player** class,

Method Name	Description
static Map<String,Integer> calculateNationalityCount(List<Player> list)	This method accepts a list of Player as argument and returns a TreeMap with the nationality as key and number of players playing for the country as value and returns the map.

In the TreeMap have the nationality as key and Count the number of players playing for the country and keep the number of players as value. Print the value sorted by country name.

The player details should be given as a comma separated value in the below order, name, dateOfBirth, skill, numberOfMatches, runs, wickets, nationality, powerRating

Print format:

```
System.out.format("%-15s %s\n","Country","Count");
```

Sample Input and Output 1:

Enter the number of players:

5

MS Dhoni,07-07-1981,Batsman,159,3561,0,India,4.4

Virat Kohli,05-11-1988,Batsman,149,4418,4,India,4.7

Bhuvneshwar,05-02-1990,Bowler,90,158,111,India,4.1

Mike hussey,27-05-1975,Batsman,59,1977,0,Australia,4.5

Mitchell Stark,30-01-1990,Bowler,27,96,34,Australia,4

Country	Count
---------	-------

Australia	2
-----------	---

India	3
-------	---

In this requirement, you need to find the country from which the maximum number of overseas players is taking part in the IPL.

a) Create a Class **Player** with the following attributes:

Member Field Name	Type
name	String
dateOfBirth	java.util.Date / LocalDate
skill	String
numberOfMatches	Integer
runs	Integer
wickets	Integer
nationality	String
powerRating	Double

Mark all the attributes as private, Create / Generate appropriate Getters & Setters, Add a default constructor and a parameterized constructor to take in all attributes in the given order: **public Player(String name, java.util.Date dateOfBirth, String skill, Integer numberOfMatches, Integer runs, Integer wickets, String nationality, Double powerRating)**

b) Create the following static method in Player class,

Method Name	Description
public static Player createPlayer(String detail)	This method accepts a string which contains player details separated by commas. Split the details and create a player object from the details and return the player object.
public static String highestCount(List<Player> playerList)	This accepts a list as argument and returns a String. It takes a player list as argument and returns the country from which the maximum number of players are playing.

The player details should be given as a comma-separated value in the below order, **name,dateOfBirth,skill,numberOfMatches,runs,wickets,nationality,powerRating**

Create a driver class Main with the main method to get details and display details.

Sample Input and Output:

Enter the number of players:

6

MS Dhoni,07-07-1981,Batsman,159,3561,0,India,4.4

Virat Kohli,05-11-1988,Batsman,149,4418,4,India,4.7

Bhuvneshwar,05-02-1990,Bowler,90,158,111,India,4.1

Ab de Villiers,7-02-1984,Batsman,129,3473,0,S Africa,4.7

Mitchell Stark,30-01-1990,Bowler,27,96,34,Australia,4.1

Faf du Plessis,13-07-1984,Batsman,53,1295,0,S Africa,4.5

The nationality with maximum players:India