Line Up

This problem is worth 15 points. One point for the student tester (given with the folder), one point for correctly implementing the four methods in the `Player` class, three points for each of the four methods in the `LineUp` class, and one bonus point for successfully earning all 14 point.

Problem Description: You work for the IT department of a baseball team. Baseball has an almost infinite number of stats—what a great job for a computer programmer! You will write a program to determine the first four spots in the batting order given a List of Players with their stats.

In this problem you are given an incomplete immutable helper class `Player` that is used to represent different players on a baseball team. The `Player` class has 8 completed accessor methods which report the Players name, number of at bats, number of hits, number of doubles, number of triples, number home runs, and the number of base on balls or walks. You must complete four additional accessor methods.

The header for the Player constructor is:

```
public Player(  String playerName,    /*  Players Name  */
                int ab,          /*  number of at bats  */
                int hits,        /*  number of hits */
                int doubles,     /*  number of doubles */
                int triples,     /*  number of triples */
                int hrs,         /*  number of home runs */
                int bbs          /*  number of base on balls or walks */)
```

note: definition of terms are given as needed

The first method in the `Player` class you must complete is the `int getNumSingles()`. `getNumSingles` returns the number of singles for this player. The number of singles is determined by taking the number of hits and subtracting the number of doubles and subtracting the number of triples and subtracting the number of home runs.

The following code shows the results of the `getNumSingles` method.

The following code	Returns
`Player seager = new Player("Seager", 150, 50, 9, 2, 6, 8);`	
`seager.getNumSingles();`	`33 = 50-9-2-6`

The second method in the `Player` class you must complete is the `double getSluggingPercent()`. `getSluggingPercent` returns the total bases divide by the number of at bats. Total bases is calculated by adding the number of singles, two times the number of doubles, three times the number of triples and four times the number of home runs.

The following code shows the results of the `getSluggingPercent` method.

The following code	Returns
`Player seager = new Player("Seager", 150, 50, 9, 2, 6, 8);`	
`seager.getSluggingPercent ();`	`81/150. = (9*2 + 2*3 + 6*4 + 50-9-2-6) / 150.`

Borrowed (and modified) from: Pacific Lutheran University - February 2014, Programming Contest

The third method in the `Player` class you must complete is the `double getBattingAverage()`. `getBattingAverage` returns the total number of hits divide by the number of at bats.
In case you are wondering, base on balls do not count as at bats when calculating batting average.

The following code shows the results of the `getBattingAverage` method.

The following code	Returns
`Player seager = new Player("Seager", 150, 50, 9, 2, 6, 8);`	
`seager.getBattingAverage();`	$0.3\bar{3} = 50/150.$

The fourth method in the `Player` class you must complete is the `double getOnBasePercentage()`. `getOnBasePercentage` returns the total number of hits and base on balls (or walks) divide by the total number of at bats and base on balls.

The following code shows the results of the `getOnBasePercentage` method.

The following code	Returns
`Player seager = new Player("Seager", 150, 50, 9, 2, 6, 8);`	
`seager.getOnBasePercentage();`	`0.367088…. = ((50+8)/(150+8.)`

You will complete four methods in the `LineUp` class. The `LineUp` class has a single constructor which creates the instance variable `myTeam`, an `ArrayList<Player>` which is used to store the nine players from which the line is generated. The `LineUp` class has the completed helper method `void addPlayers(List<Player> ps)` which copies the players from `ps` to the instance variable `myTeam`.

The first method you are to complete is `void getCleanUp()`. `getCleanUp` selects the `Player` that is to occupy the fourth spot in the batting order and is the `Player` with the highest slugging percentage. In case of a tie, select the Player with most home runs. If there is still a tie select the Player with greatest Batting average. You may assume there will not be a tie after selecting the Player with greatest Batting average.

The following code shows the results of the `getCleanUp` method.

The following code	Returns

`LineUp lu = new LineUp();`
`List<Player> ps = new ArrayList<Player>();`
 	
`ps.add(new Player("Seager", 150, 50, 9, 2, 6, 8));`
`ps.add(new Player("Ruth", 135, 52, 10, 1, 17, 12));`
`ps.add(new Player("Trout", 145, 40, 18, 5, 13, 9));`
`ps.add(new Player("Beltre", 175, 45, 3, 0, 1, 2));`
`ps.add(new Player("Alltuve", 125, 72, 23, 5, 10, 11));`
`ps.add(new Player("Bellinger", 150, 40, 11, 0, 14, 7));`
`ps.add(new Player("Puig", 145, 40, 19, 4, 13, 9));`
`ps.add(new Player("Ohtani", 115, 25, 2, 0, 6, 4));`
`ps.add(new Player("Arenado", 155, 51, 19, 0, 16, 10));`
`lu.addPlayers(ps);`	

`lu.getCleanUp());`	`new Player("Alltuve", 125, 72, 23, 5, 10, 11)`

* Alltuve has the slugging percent of (23*2+5*3+10*4+(72-23-5-10)) / 125

Borrowed (and modified) from: Pacific Lutheran University - February 2014, Programming Contest

The second method you are to complete is `void getLeadOff()`. `getLeadOff` selects the `Player` that is to occupy the first spot in the batting order. The `Player` that is to occupy the first spot must not be the Clean Up batter and is the `Player` with highest on Base Percentage. In case of tie, select the Player with greatest Batting average. If there is still a tie select the Player with most singles. You may assume there will not be a tie after selecting the Player with greatest Batting average.

The following code shows the results of the `getLeadOff` method.

The following code	Returns

`LineUp lu = new LineUp();`
`List<Player> ps = new ArrayList<Player>();`
 	
`ps.add(new Player("Seager", 150, 50, 9, 2, 6, 8));`
`ps.add(new Player("Ruth", 135, 52, 10, 1, 17, 12));`
`ps.add(new Player("Trout", 145, 40, 18, 5, 13, 9));`
`ps.add(new Player("Beltre", 175, 45, 3, 0, 1, 2));`
`ps.add(new Player("Alltuve", 125, 72, 23, 5, 10, 11));`
`ps.add(new Player("Bellinger", 150, 40, 11, 0, 14, 7));`
`ps.add(new Player("Puig", 145, 40, 19, 4, 13, 9));`
`ps.add(new Player("Ohtani", 115, 25, 2, 0, 6, 4));`
`ps.add(new Player("Arenado", 155, 51, 19, 0, 16, 10));`
`lu.addPlayers(ps);`
`lu.getCleanUp());`	

`lu.getLeadOff();`	`new Player("Ruth", 135, 52, 10, 1, 17, 12)`

* Alltuve is the clean up hitter and not available as the leadoff, therefore Ruth has a on base percentage of (52+12) / (135+12)

The third method you are to complete is `void getThirdBatter()`. `getThirdBatter` selects the `Player` that is to occupy the third spot in the batting order. The `Player` that is to occupy the third spot must not be the Leadoff batter and must not be the Clean Up batter and is the `Player` with the greatest number of hits. In case of tie, pick the Player most number of At Bats, if still tied pick the Player with the most doubles plus triples.

The following code shows the results of the `getThirdBatter` method.

The following code	Returns

`LineUp lu = new LineUp();`
`List<Player> ps = new ArrayList<Player>();`
 	
`ps.add(new Player("Seager", 150, 50, 9, 2, 6, 8));`
`ps.add(new Player("Ruth", 135, 52, 10, 1, 17, 12));`
`ps.add(new Player("Trout", 145, 40, 18, 5, 13, 9));`
`ps.add(new Player("Beltre", 175, 45, 3, 0, 1, 2));`
`ps.add(new Player("Alltuve", 125, 72, 23, 5, 10, 11));`
`ps.add(new Player("Bellinger", 150, 40, 11, 0, 14, 7));`
`ps.add(new Player("Puig", 145, 40, 19, 4, 13, 9));`
`ps.add(new Player("Ohtani", 115, 25, 2, 0, 6, 4));`
`ps.add(new Player("Arenado", 155, 51, 19, 0, 16, 10));`
`lu.addPlayers(ps);`
`lu.getCleanUp());`	

`lu.getThirdBatter();`	`new Player("Arenado", 155, 51, 19, 0, 16, 10)`

* Alltuve is the clean up hitter and Ruth is the leadoff batter and both are not available as the third batter, therefore Arenado has the greatest number of hits, 51.

The fourth method you are to complete is `void getSecondBatter()`. `getSecondBatter` selects the `Player` that is to occupy the second spot in the batting order. The `Player` that is to occupy the second spot must not be the third batter, Leadoff batter or the Clean Up batter and is the `Player` with the most base on balls. In case of tie, pick the Player with highest on Base Percentage, if still tied pick Player with most triples.

The following code shows the results of the `getSecondBatter` method.

The following code	Returns

`LineUp lu = new LineUp();`
`List<Player> ps = new ArrayList<Player>();`
 	
`ps.add(new Player("Seager", 150, 50, 9, 2, 6, 8));`
`ps.add(new Player("Ruth", 135, 52, 10, 1, 17, 12));`
`ps.add(new Player("Trout", 145, 40, 18, 5, 13, 9));`
`ps.add(new Player("Beltre", 175, 45, 3, 0, 1, 2));`
`ps.add(new Player("Alltuve", 125, 72, 23, 5, 10, 11));`
`ps.add(new Player("Bellinger", 150, 40, 11, 0, 14, 7));`
`ps.add(new Player("Puig", 145, 40, 19, 4, 13, 9));`
`ps.add(new Player("Ohtani", 115, 25, 2, 0, 6, 4));`
`ps.add(new Player("Arenado", 155, 51, 19, 0, 16, 10));`
`lu.addPlayers(ps);`
`lu.getCleanUp());`	

`lu.getSecondBatter();`	`new Player("Trout", 145,` `40, 18, 5, 13, 9)`

* Alltuve is the clean up hitter, Ruth is the leadoff batter and Arenado is the third batter and all are not available as the second batter,
Trout and Puig are both tied with the highest number of base on balls, 9
and both are tied with the same on base percentage
therefore Trout has more triples making him the second batter