APCS :: Basketball Stats Program part I

You should now have enough tools in your tool belt to get some work done on the Stats program.

Before you begin make sure your program has the following classes:

- **Player** – maintains individual player stats
- **Team** – maintains a list of Players and a score
- **Game** – maintains references to two teams, designated as “home” and “Visitor” respectively.
- **Main** – runs the UI and parses commands.

Here is a typical box-score for one team showing stats for a game. We will use this as our guide as we proceed for this part of the assignment we will only concern ourselves with fga, fgm and pts.

<table>
<thead>
<tr>
<th>PLAYER</th>
<th>POS</th>
<th>MIN</th>
<th>FGM-A</th>
<th>3GM-A</th>
<th>FTM-A</th>
<th>REBOUNDS</th>
<th>OFF</th>
<th>DEF</th>
<th>TOT</th>
<th>A</th>
<th>PF</th>
<th>STL</th>
<th>TO</th>
<th>BLK</th>
<th>PTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ben Gordon</td>
<td>G</td>
<td>40:26</td>
<td>8-17</td>
<td>0-2</td>
<td>4-4</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Kirk Hinrich</td>
<td>C</td>
<td>24:37</td>
<td>3-11</td>
<td>0-2</td>
<td>2-2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Andres Nocioni</td>
<td>F</td>
<td>31:16</td>
<td>5-14</td>
<td>2-5</td>
<td>2-4</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adrian Griffin</td>
<td>F</td>
<td>12:57</td>
<td>0-3</td>
<td>0-0</td>
<td>0-0</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Ben Wallace</td>
<td>C</td>
<td>27:47</td>
<td>2-4</td>
<td>0-0</td>
<td>1-2</td>
<td>2</td>
<td>6</td>
<td>8</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thabo Sefolosha</td>
<td></td>
<td>24:40</td>
<td>4-8</td>
<td>1-2</td>
<td>0-1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Chris Duhon</td>
<td></td>
<td>23:30</td>
<td>3-11</td>
<td>2-7</td>
<td>0-0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Joe Smith</td>
<td></td>
<td>18:00</td>
<td>2-6</td>
<td>0-0</td>
<td>0-0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Tyus Thomas</td>
<td></td>
<td>13:42</td>
<td>1-6</td>
<td>0-0</td>
<td>3-3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Thomas Gardner</td>
<td></td>
<td>11:32</td>
<td>1-3</td>
<td>0-2</td>
<td>0-0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Joakim Noah</td>
<td></td>
<td>5:51</td>
<td>0-1</td>
<td>0-0</td>
<td>0-0</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Aaron Gray</td>
<td></td>
<td>5:51</td>
<td>2-5</td>
<td>0-0</td>
<td>0-0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Please implement the following commands and test them to make sure they work. You may hard-code the building of teams in main in order to test.

**Command: fga**

**Description:**

- **fga** *H|V* *num* *pts*

Records a field goal attempt for the player indicated.

- “H” or “V” indicates Home or Visitor team
- num indicates player number – MUST exist on roster of team
- pts indicates points scored – MUST be 0 or 2
Print an error message to the screen (don’t throw an exception) if:

- The command is not understood for some reason.
- The player with num does not exist on the team given

Example:
fga H 21 2

Records a field goal attempt for player with number 21 on the “Home” team – she scored 2 points so the field goal was made. You may want to print a message indicating a successful record was made.

Command: **print score|team [H|A]**

Description:
print score (This is the first option of many things that we will eventually print)

Prints the current game score to the console. The game score should be an accumulation of points for each team as recorded from calls to fga.

EXAMPLE:
$> print score
Home: 30
Visitor: 23
$>

NOTE: You may print the actual team names instead of “Home” and “Visitor” if you like.

Description: print team H|A
This call to print will print a summary of player statistics for the team indicated.

EXAMPLE:
$> print team H
Team: foobars
11 Smith 3-4 6
12 McCormick 5-8 10
14 Davis 1-5 2
$>

The above example shows a space-delimited record for each player with the following fields:
num name fgm-fga pts
Recommendation:
Make each line of stats be the result of the toString method from the Player class. This will make your life much easier in the future.

Print an error message to the screen if:
   The argument given to print is not understood or is incomplete.
Part II: Reorganizing the program into a manageable, extensible structure:

Command Class
Command Factory
Part III: Go to town with commands!

Below is a long list of potential commands that you could write. Some will be very easy to add, others more challenging.

YOU DO NOT have to use the command names that I suggest, or even the argument lists. My intent in describing the commands below is to give some sense of the variety of commands that you might want to implement a good basketball statistics program.

FOR EXAMPLE: all of the game play commands listed below have an “HorV” argument so that the user can indicate which team the player belongs to. This could become cumbersome to type. You are free to change the argument to be a lower-case “h” or “v” instead of upper case. Or you could implement two different commands: fga and FGA where the lowercase version means one team and the uppercase version means the other.

There are two categories of commands:

• Administrative Commands – commands that help in setting up the program to record and report statistics about a game. These commands will tend to involve more processing and will take more time to implement. They are also, for the most part, more interesting.

• Game Play Commands – commands that are to be used during live play. A premium on speed of entry should be given to these commands since you’ll have to enter them quickly.

### Administrative Commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>PURPOSE</th>
<th>USAGE</th>
<th>ARGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ng /new game</td>
<td>Start collecting stats for a new game. You can optionally include a lot of information through arguments that you would want to set otherwise. This command will evolve and may become quite complex. For example, what if this command is issued in the middle of a game? You might want to make sure that all of the data isn’t wiped out – ask they user if they’re sure.</td>
<td>ng [date] [location] [teamH] [teamV]</td>
<td>[date] – Optional. The date the game took place. [location] – Optional. Where the game took place. [teamH] – The name of the home team OR the a path to a text file with information about the home team. [teamV] – The name of the visiting team OR a path to a text file with its information.</td>
</tr>
</tbody>
</table>

**EXAMPLES:**

ng
Start a new game with everything empty. User will have to load or set all game information.

ng 11.21.2007 UHigh-Tourney
Start a new game setting the date of the contest and the place it occurred.

```
ng 11.21.2007 UHigh maroonsRoster.txt latinRoster.txt
```
Start a new game with the date, location, and create two teams using the text files provided.

---

**save game / sg**

**PURPOSE**
Save the game to a text file. This could be imagined as saving every command that was executed as plain text to a file. In theory, this file could be read back into the program (see load game) to regenerate game statistics etc. i.e. save a log of every command that was issued and executed. There are other ways you could save the game as well that might allow for more efficient viewing and loading.

**USAGE**
```
sg filename
save game filename
```

**ARGS**
- `fileName` – the path/to/a/text/file.txt to which game data will be written.

**EXAMPLES:**
```
sg Documents/games/uhighVLatin.txt
```
Saves a game log to the text file specified.
```
save game Documents/games/uhighVLatin.txt
```
Same as above more verbosely.

---

**exec / e**

**PURPOSE**
Execute commands one by one that are listed in a plain text file. This would allow you to write a simple file to either get a game started (loading rosters, setting home, visitor, etc.) or by mere extension execute any list of commands you provide. This would be a very handy command to have around for testing purposes as well.

**USAGE**
```
exec filename
```

**ARGS**
- `fileName` – the path/to/a/text/file.txt which contains commands for your program, one to a line.

**EXAMPLES:**
```
exec uhighVLatin.txt
```
Opens the text file and interprets each line as though it were a line entered at the keyboard.

---

**print / p**

**PURPOSE**
Print a pre-defined report. You may want to break this up into different commands rather than trying to fit it all into one.

**USAGE**
```
print (box|player|playbyplay) [HorV] [num]
```
ARGS

box|player|playbyplay

These are three named reports that could be printed.

[HorV] – required by Box and Player reports respectively. To print a box score for a team you need to know which team. If left blank print the box score for both teams. If you’re going to print a Player’s stats you need to know which team.


EXAMPLES:

print box H
Print the box score for the home team.

print box
Print the box scores for both teams.

print playbyplay
Print a play-by-play listing of everything that happened in the game. This should be a human readable transcript of the game.

Print Player H 21
Print the individual stats for player 21 on the home team. This could be virtually the same as the home team box score but just showing one player.

---

load game / lg

PURPOSE
Load an entire game from a plain text file of commands to be issued. The idea would be that you could load an entire game from the commands that were issued.

USAGE

sg filename
save game filename

ARGS

fileName – the path/to/a/text/file.txt to which game data will be written.

EXAMPLES:

sg Documents/games/uhighVLatin.txt
Saves a game log to the text file specified.

save game Documents/games/uhighVLatin.txt
Same as above more verbosely.

---

lt / load team

PURPOSE
Create and load players into a team from a data file for the current game.

USAGE

lt HorV fileName
load team HorV filename

ARGS

HorV -- A single character “H” or “V” (not necessarily upper case) that represents whether the team to load is the home or visiting team.

fileName – the path/to/a/text/file.txt that holds team information formatted in a specified way.

EXAMPLES: lt H Documents/rosters/uhigh.txt
This command should load the data contained in uhigh.txt into a new Team specified as the home team.

---

**add**

**PURPOSE**
Add a team to the game or a player to a roster.

**USAGE**
`add (team|player) HorV name [number]`

**ARGS**
- `team|player` – are you adding a team or a player?
- `HorV` – a single character “H” or “V” is the team (or player on) the home team or visiting team?
- `name` – the name of the team or of the player being added.
- `[number]` – required for adding a player, omitted for a team.

**EXAMPLES:**
- **add team H Foobars**
  Add a team called “Foobars” to the game as the home team.
- **add player H Baker 14**
  Add a player to the home team roster with the name “Baker” and number 14.

---

**undo**

**PURPOSE**
Undo the last command that was executed. (Note that this requires that you keep a log of the commands that were executed. I’d recommend an array of Commands). Multiple calls to undo should set back through history. Undoing a command should reset player/team/game data to the state it was in before the command was executed, and it should remove the command from the list.

**USAGE**
`undo [num]`

**ARGS**
- `num` – optional the number of commands to undo. i.e. go back in history undoing num number of commands.

**EXAMPLES:**
- **undo**
  Undoes the last command that was executed.
- **undo 14**
  Undoes the last 14 commands

---

**Game Stats:**

---

**fga**

**PURPOSE**
Record a “Field Goal Attempt” for a player on some team. The command should update team and player statistics accordingly.

**USAGE**
`fga HorV num pts`

**ARGS**
- `HorV` -- A single character “H” or “V” (not necessarily upper case) that represents whether the player who took the
shot is on the “Home” or “Visiting” team.

num – the number of the player who took the shot

pts – The number of points scored by the player when they took the shot: 0 indicates a miss, 2 indicates a make. 1 is not valid because that would come from a free throw attempt (see fta). NOTE: 3 is a valid value only if you don’t plan to implement the 3pt command – which is okay.

EXAMPLES:

fga H 21 2
This would indicate that player with number 21 for the home team took a shot and scored two points. The player’s fga and fgm stats should be updated, as well as the players total points. The Home team should also have two points recorded for the team score.

fga H 21 0
Would indicate that player 21 on the home team took a shot and missed.

reb

PURPOSE
Record a “Rebound” for a player on some team. The command should update team and player statistics accordingly.

USAGE
reb HorV num [OorD]

ARGS
HorV -- A single character “H” or “V” that represents whether the player who got the rebound is on the “Home” or “Visiting” team.
num – the number of the player who got the rebound

[OorD] – OPTIONAL a single character “O” or “D” to indicate an offensive or defensive rebound.

EXAMPLES:

reb H 21
This would indicate that player with number 21 for the home team took got a rebound. The player’s total rebounds should be incremented. Typically, team rebounds are statistic that’s calculated when the game is over, but you can update this team stat in real time if you like – just like points.

reb H 21 D
Would indicate that player 21 on the home team got a “Defensive” rebound.

fta

PURPOSE
Record a “Free Throw Attempt” for a player on some team. The command should update team and player statistics accordingly.

USAGE
fta HorV num pts

ARGS
HorV -- A single character “H” or “V” that represents whether the player who took the shot is on the “Home” or “Visiting” team.
num – the number of the player who took the shot
pts – The number of points scored by the player when they
took the shot: 0 indicates a miss, 1 indicates a make. Any other values are not valid for free throws.

**EXAMPLES:**

`fta H 21 0`
This would indicate that player with number 21 for the home team shot a free three and missed. The player’s fta stats should be updated accordingly, similar to fga.

---

**a**

**PURPOSE**
Record an “assist” for a player on some team. The command should update team and player statistics accordingly. (An assist is a pass that leads to a made basket – this is a subjective stat determined by the scorekeeper. Rule of thumb: if the player who received the pass does not dribble before taking the shot, the player who made the pass gets an assist).

**USAGE**

```
a HorV num
```

**ARGS**

- `HorV` -- A single character “H” or “V” that represents whether the player who made the pass is on the “Home” or “Visiting” team.
- `num` – the number of the player who gets the assist

**EXAMPLES:**

`a V 21`
Player 21 on the Visiting team gets an assist.

---

**set quarter / sq**

**PURPOSE**
Set the game quarter (Basketball is played in 4 quarters. Game statistics are often broken down by which quarter certain things occurred. A very common thing to look at after the game is scoring by quarter. For example, You might see that the game was won or lost in the 3rd quarter when the home team scored 20 points while visitor only scored 5).

**USAGE**

```
sq num
```

**ARGS**

- `num` – the quarter number. 1,2,3 or 4 are valid, as is “OT.” Thus, num should probably be kept as a string. If the game is tied after four quarters the game goes to “Overtime” OT. Furthermore you can have multiple OT periods. OT1, OT2, etc.

**EXAMPLES:**

`sq 1`
`sq OT2`

---

**pf / foul**

**PURPOSE**
Log a “Player Foul.” Each player has 5 fouls to give during the course of a basketball game – on the 5th foul the player must leave the game and cannot check back in. Team fouls
Every foul charged to a player increments the “team fouls” by 1. When a team A accumulates a certain number of fouls (7 in High School basketball), team B gets to shoot free throws for each subsequent foul.

**Usage**

\[ pf \ HorV \ num \]

**Args**

- **HorV** -- A single character “H” or “V” (not necessarily upper case) that represents which team the player plays for
- **num** -- the player’s number

**Examples:**

`pf H 21`

Record a foul for player 21 on the home team. This should also increase team fouls by 1.

---

**Purpose**

Record a steal for a player. (A player steals the ball when he or she gains possession of the ball from the other team in a way that is not a rebound. The most common form of steal is an intercepted pass).

**Usage**

\[ stl \ HorV \ num \]

**Args**

- **HorV** -- A single character “H” or “V” (not necessarily upper case) that represents which team the player is on.
- **num** -- the player’s number

**Examples:**

`stl H 21`

Record a steal for player 21 on the home team.

---

**Purpose**

Record a turnover for a player. (A player “turns the ball over” when he or she does something to allow the other team to gain possession of the ball without having taken a shot. For example, if a player on team A gets a steal, it means that a player on team B committed a turnover. There are a number of violations that result in turnovers as well – traveling, double dribble, etc. Losing the ball out of bounds is another form of turnover).

**Usage**

\[ stl \ HorV \ num \]

**Args**

- **HorV** -- A single character “H” or “V” (not necessarily upper case) that represents which team the player is on.
- **num** -- the player’s number

**Examples:**

`stl H 21`

Record a steal for player 21 on the home team.

---

**Purpose**

Record a “blocked shot” for a player. Player A gets a block
when he or she deflects the ball being shot by another player.

<table>
<thead>
<tr>
<th>USAGE</th>
<th>blk HorV num</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARGS</td>
<td>HorV -- A single character “H” or “V” (not necessarily upper case) that represents which team the player is on num -- the player’s number</td>
</tr>
<tr>
<td>EXAMPLES</td>
<td>blk H 21</td>
</tr>
<tr>
<td></td>
<td>Record a block for player 21 on the home team.</td>
</tr>
</tbody>
</table>

---

**3pt / three point shot**

<table>
<thead>
<tr>
<th>PURPOSE</th>
<th>Record a three-point shot attempt for a player. This is exactly the same as the fga command but specific to 3-point shots.</th>
</tr>
</thead>
<tbody>
<tr>
<td>USAGE</td>
<td>3pt HorV num pts</td>
</tr>
<tr>
<td>ARGS</td>
<td>HorV -- A single character “H” or “V” (not necessarily upper case) that represents which team the player is on num -- the player’s number pts -- the number of points scored as a result of the shot attempt – 0 or 3 are valid values.</td>
</tr>
<tr>
<td>EXAMPLES</td>
<td>3pt H 21 3</td>
</tr>
<tr>
<td></td>
<td>Record a 3pt shot attempt and make for player 21 on the home team. Player 21’s points should be increased by 3 as well as the team points.</td>
</tr>
</tbody>
</table>

---

**c / comment**

<table>
<thead>
<tr>
<th>PURPOSE</th>
<th>Add a “scorekeepers comment” to the game log. Often, things happen in a game to which normal statistics simply don’t apply. Beyond bizarre circumstances, notes in the game log often help to reconstruct the game after the game is over. See examples.</th>
</tr>
</thead>
<tbody>
<tr>
<td>USAGE</td>
<td>c comment_string</td>
</tr>
<tr>
<td>ARGS</td>
<td>comment_string -- A single line of text that will be inserted into the game log.</td>
</tr>
<tr>
<td>EXAMPLES</td>
<td>c “Player number 21 injured on last play – left game on a stretcher.”</td>
</tr>
<tr>
<td></td>
<td>c “Game stopped due to leaky roof. Took 45 minutes to fix before game resumed.”</td>
</tr>
</tbody>
</table>