PART 1. The Middle

Each line of a box score can be one of four kinds of stats: Passing, Rushing, Receiving, or Kicking, as shown below:

Player ID	**Compl/Att**	**Yds**	**TDs**	**Ints**
2 Doug Flutie	1/3	14	0	1
Player ID	**Rushes**	**Yds**	**Avg**	**TDs**
2 Doug Flutie	3	13	4.3	1
Player ID	**Rec.**	**Yds**	**Avg**	**TDs**
2 Doug Flutie	1	9	0.0	0
Player ID	**FG**	**Pctg**	**Long**	**Pts**
2 Doug Flutie	0	.000	-	1

In your test plan, most of your test cases will have one line of box score data as input, and the expected number of points as expected output. For example:

Test Case 1

Input:

Player ID	Stat Type	Compl/Att	Yds	TDs	Ints
2 Doug Flutie	Passing	1/3	14	0	1

Expected Output: 0 + 0 + 0 – 0.5 = **-0.5 pts**

Test Case 2

Input:

Player ID	Stat Type	Rushes	Yds	Avg	TDs
2 Doug Flutie	Rushing	3	13	4.3	1

Expected Output: 1.3 + 6 + 0 = **7.3 pts**

Test Case 3

Input:

Player ID	Stat Type	Rec.	Yds	Avg	TDs
2 Doug Flutie	Receiving	1	9	9.0	0

Expected Output: 0.9 + 0 + 0 = **0.9 pts**

Test Case 4

Input:

Player ID	Stat Type	FG	Pctg	Long	Pts
2 Doug Flutie	Kicking	0/0	.000	-	1

Expected Output: 0 + 1 + 0 + 0 = **1 pts**

Note: These were taken from the box score example at the top of this page; your test plan might use different numbers to do better boundary testing.

Most iterations of your TDD development will add a little bit of scoring functionality to your software. Your test cases might need to be broken down further to accommodate this incremental functionality. For example, many students showed a TDD plan that looked something like this:

Iteration	Functionality	Test Case
15	Add points for successful field goals (3 pts each)	12
16	Deduct points for missed field goals (-0.4 pts each)	12
17	Add points for successful extra points (1 pt each)	12
18	And bonus points if longest field goal is 50+ yards (2 pts)	12

where Test Case 12 looked like this:

Test Case 12

Input:

Player ID	Stat Type	FG	Pctg	Long	Pts
2 Doug Flutie	Kicking	3/4	.750	41	10

Expected Output: 9 + 1 – 0.4 + 0 = **9.6 pts**

However, this test case won't work for Iterations 15, 16, and 17! After we add the code for Iteration 15, the kicker will only be rewarded 9 points for the three successful field goals, because the other functionality has not been coded yet.

There are two ways we can address this. The first is to create multiple test cases that add new functionality in succession. For example:

Test Case 12 (test for successful field goals)

Input:

Player ID	Stat Type	FG	Pctg	Long	Pts
2 Doug Flutie	Kicking	3/3	1.000	49	9

Expected Output: 9 pts

Test Case 13 (add test for missed field goals)

Input:

Player ID	Stat Type	FG	Pctg	Long	Pts
2 Doug Flutie	Kicking	3/4	.750	49	9

Expected Output: 9 – 0.4 = **8.6 pts**

Test Case 14 (add test for successful extra points)

Input:

Player ID	Stat Type	FG	Pctg	Long	Pts
2 Doug Flutie	Kicking	3/5	.600	49	11

Expected Output: 9 – 0.8 + 2 = **10.2 pts**

Test Case 15 (add test for long FG bonus)

Input:

Player ID	Stat Type	FG	Pctg	Long	Pts
2 Doug Flutie	Kicking	3/5	.600	53	11

Expected Output: 9 – 0.8 + 2 + 2 = **12.2 pts**

A second way to address this is by using a single test case, but adjusting the expected output for each iteration:

Test Case 12

Input:

Player ID	Stat Type	FG	Pctg	Long	Pts
2 Doug Flutie	Kicking	3/4	.750	53	10

Expected Output: 9 – 0.4 + 2 + 1 = **11.6 pts**

Expected Output after I-15 (3 pts/FG)**: 9 pts**
Expected Output after I-16 (missed FG deductions)**:** 9 – 0.4 = **8.6 pts**
Expected Output after I-17 (1 pt/XP)**:** 9 – 0.4 + 1 = **9.6 pts**
Expected Output after I-18 (2 bonus pts for long FG 50+ yds)**:** 9 – 0.4 + 1 + 2 = **11.6 pts**

Part 2. The Beginning

A good place to start is making sure that your program can read and parse a box score file. I am including one from an NFL game. Your first test case might look something like this:

Test Case 1 (reading input file)

Input: RAMS SEAHAWKS Boxscore.rtf (filename)

Expected Output: The program will print a table with all stats for this game; it should look like this:

```
PASSING
PLAYER                        C/ATT       YDS         TD          INT
SEAHAWKS 3 RUSSELL WILSON     11/16       152         1           1
SEAHAWKS 7 GENO SMITH         10/17       131         1           1
RAMS 9 MATTHEW STAFFORD       25/37       365         1           1

RUSHING
PLAYER                        CAR         YDS         AVG         TD
SEAHAWKS 41 ALEX COLLINS      15          47          3.1         0
SEAHAWKS 7 GENO SMITH         3           28          7.7         0
SEAHAWKS 3 RUSSELL WILSON     2           10          5.0         0
SEAHAWKS 31 DEEJAY DALLAS     4           7           1.8         0
RAMS 27 DARRELL HENDERSON JR. 17          82          4.8         1
RAMS 25 SONY MICHEL           11          37          3.4         1
RAMS 9 MATTHEW STAFFORD       1           -1          -1.0        0
```

```
RECEIVING                     REC         YDS         AVG         TD
SEAHAWKS 14 DK METCALF        5           98          19.6        2
SEAHAWKS 19 TYLER LOCKETT     5           57          11.4        0
SEAHAWKS 31 DEEKAY DALLAS     2           32          16.0        0
SEAHAWKS 89 WALT DISSLY       2           29          14.5        0
RAMS 2 ROBERT WOODS           12          150         12.5        0
RAMS 10 COOPER KUPP           7           92          13.1        0
RAMS 1 DESEAN JACKSON         1           68          68.0        0
RAMS 27 DARRELL HENDERSON JR. 1           17          17.0        0
RAMS 12 VAN JEFFERSON JR.     1           16          16.0        0
RAMS 89 TYLER HIGBEE          2           14          7.0         1
RAMS 25 SONY MICHEL           1           8           8.0         0

KICKING                       FG          PCT         LNG         PTS
SEAHAWKS 5 JASON MYERS        1/2         .500        32          5
RAMS 8 MATT GAY               2/2         1.000       47          8
```

Note: The original file didn’t look like this; it looks like this:

```
SEAHAWKS
PASSING
PLAYER                   C/ATT     YDS       TD        INT
3 RUSSELL WILSON         11/16     152       1         1
7 GENO SMITH             10/17     131       1         1

RUSHING
PLAYER                   CAR       YDS       AVG       TD
41 ALEX COLLINS          15        47        3.1       0
7 GENO SMITH             3         28        7.7       0
3 RUSSELL WILSON         2         10        5.0       0
31 DEEJAY DALLAS         4         7         1.8       0

RECEIVING                REC       YDS       AVG       TD
14 DK METCALF            5         98        19.6      2
      :                  :          :         :        :
```

The software has done two things to the input file:

1. It has added the team name to the player ID, so that each player can be identified with three fields:
   team name, jersey number, and name.
2. It has grouped the passers, rushers, receivers, and kickers from both teams in the same part of the table.

This was never a specified requirement, but you may find it easier to process a box score file if you do some preprocessing first. This preprocessing should be part of your TDD development, where you develop the test cases before your write your code.

Part 3. The End

Your program needs to be able to add together all points for a player. For example, in the game in this example, Geno Smith will receive both passing points and rushing points. So, you might have three test cases to ensure this happens:

Test Case 9 (checking passing stats)

Input:

Player ID	Stat	C/Att	Yds	TD	Int
Seahawks 7 Geno Smith	Passing	10/17	131	1	1

Expected Output: 2 + 6 + 4.5 – 0.5 = **12 pts**

Test Case 19 (checking rushing stats)

Input:

Player ID	Stat	Car	Yds	Avg	TD
Seahawks 7 Geno Smith	Rushing	3	28	7.7	0

Expected Output: 2.8 + 2 + 0 = **4.8 pts**

Test Case 27 (checking total stats)

Input:

Player ID	Stat	C/Att	Yds	TD	Int
Seahawks 7 Geno Smith	Passing	10/17	131	1	1
	Stat	**Car**	**Yds**	**Avg**	**TD**
Seahawks 7 Geno Smith	Rushing	3	28	7.7	0

Expected Output: 2 + 6 + 4.5 – 0.5
+ 2.8 + 2 + 0 = **16.8 pts**