*Turn on the Navigation Pane in Word when viewing this document:* **View tab > enable Navigation pane**

# Program Specification

* *Program title:* Victorino Player Reporting System
* *File name:* MIS3301-TeamProject.py
* *Due Date:* see Canvas for due date
* Victorino Consulting is asking you to develop the Victorino Player Reporting System – a CSV file with player data has been provided in Canvas. The Player class shown here contains the attributes in this file. Also, see the section below entitled “About the Data” for more information.

# Read before starting

* Reminder: This is a *pair* project. Students are expected to complete this assignment with their partner and without the assistance of *other* individuals or pairs. If you need help, contact your instructor via email or visit during office hours.
* All of your code for this assignment must be written in functions.
* Globals - global constants are permitted but global variables are not.
* Continue to follow the same **quality standards** as before: naming of variables & constants, comment lines, leaving a blank line between major sections and new structures (if, loops, etc.), no redundant functionality, formatting output, data validation, cannot crash, alignment, formatting, good variable names, etc.
* You will be graded based on the quality of the application including the code, data entry, output, and usability. You will also be graded on whether you used of all of the major concepts we have covered – see chapter names for guidance (e.g. loops, lists, etc.)
* Your code must not have any redundant code nor any hardcoded values.
* **This code must be free of syntax errors, or it will not be graded. Thus, comment out any code that crashes, or for a better grade, improve the code and submit it one day late with a 10% deduction.**

# Project Approach

## Scrum

We are following the *Scrum* approach to development. As such, the client has provided a *Product Backlog* of *user stories* they would like for you to develop. Since your team consists of only 2 members, you will select only one user story to complete during this 10-day *sprint*.

## Pair Programming

For this project, you will employ *Pair Programming*. You will turn in only one assignment with both students’ names in the header comments. **You MUST work with someone in your section.**

* Each pair will develop code together at one computer. There will be one *driver* (typing the code) and the other is the *navigator* (pointing out issues and making suggestions). The driver is expected to “program out loud” – i.e. talk about what he/she is doing while coding. The navigator should be actively engaged as well. **The goal is to take turns being driver and navigator**.
* The benefit of this approach is that, when done correctly, it should increase code quality and knowledge transfer (both in the problem domain as well as in programming techniques). Pair Programming is a feature of XP (Extreme Programming) development and may also be used in other agile approaches such as Scrum.
* **Important:** you are to **submit the project only under one student account**. Both student names must be documented in the comments section at the top of your code.

# Instructions

## Step 1 – Add Comment Lines

* Copy the following code to your program.

|  |
| --- |
| # Include the required header comment lines at the top of your code (as before). Include your both member names that participated in the project. If a member does not participate, still include their name & write “DID NOT PARTICIPATE” next to their name.  **#import modules**  import csv  import os  import random  from operator import itemgetter  **#constants**  PLAYER\_FILE = 'NFL-players.csv'  **#Parallel Lists - Teams**  **TEAM\_CODES** = ['ATF', 'AZC', 'BFB', 'BLR', 'CAP', 'CHB', 'CLB', 'CNB', 'DLC', 'DTL',  'DVB', 'GBP', 'HST', 'INC', 'JKJ', 'KCC', 'LAC', 'LAR', 'LVR', 'MMD',  'MNV', 'NEP', 'NOS', 'NYG', 'NYJ', 'PHE', 'PTS', 'SFN', 'STS', 'TBB', 'TNT', 'WAS']  **TEAM\_NAMES** = ['Atlanta Falcons', 'Arizona Cardinals', 'Buffalo Bills', 'Baltimore Ravens', 'Carolina Panthers',  'Chicago Bears', 'Cleveland Browns', 'Cincinnati Bengals', 'Dallas Cowboys', 'Detroit Lions',  'Denver Broncos', 'Green Bay Packers', 'Houston Texans', 'Indianapolis Colts', 'Jacksonville Jaguars',  'Kansas City Chiefs', 'Los Angeles Chargers', 'Los Angeles Rams', 'Las Vegas Raiders', 'Miami Dolphins',  'Minnesota Vikings', 'New England Patriots', 'New Orleans Saints', 'New York Giants', 'New York Jets',  'Philadelphia Eagles', 'Pittsburgh Steelers', 'San Francisco 49ers', 'Seattle Seahawks', 'Tampa Bay Buccaneers',  'Tennessee Titans', 'Washington Football Team']  **#Parallel Lists - Positions**  **POS\_CODES** = ['C', 'CB', 'DE', 'DT', 'FB', 'G', 'K', 'LB', 'LS', 'NT', 'OT', 'P', 'QB', 'RB', 'S', 'TE', 'WR']  **POS\_NAMES** = ['Center', 'Cornerback', 'Defensive End', 'Defensive Tackle', 'Fullback', 'Guard', 'Kicker', 'Linebacker', 'Long Snapper', \  'Nose Tackle', 'Offensive Tackle', 'Punter', 'Quarterback', 'Running back', 'Safety', 'Tight End', 'Wide Receiver'] |

## Step 2 – Code the main() function

* Write code to open the *player* CSV file for reading.
* Display a banner with the application title: *Victorino Player Reporting System*
* Add code to display the following menu. Ensure the menu appears within a loop so that the menu is redisplayed after the user has selected an option.

1 - Team Roster Report

2 - Team Weight Analysis Report

3 - Filtered Players Report

X - Exit

## Step 3 – Add Functions for each user story

* Add function *defs* for all 3 user stories; however, you will only complete the code for one.
* You and your partner will pick only one user story to complete. Read the details for each user story below before deciding.
* For the other 2 user stories, just add one print statement in the function stating that this functionality will be completed in Release 2.0. Also, go back to the menu and annotate these 2 menu choices with the phrase " - Release 2.0" next to the option name.

## Step 4 – Complete the Functionality for your user story

* Proceed to code your user story.

# Product Backlog

Below is the Product Backlog (i.e. the list of user stories to be completed). You and your partner will select up *one* of the 3 user stories to work on and complete.

## US1 – View Team Roster Report (by Team & Position)

### User Story

|  |
| --- |
| ***User Story:*** As a manager in the league, I need to view a roster for multiple teams so that I can quickly lookup player details.  ***Acceptance Criteria:***   * + Report must display all teams as specified by user and must include the following data: team, position, last name, first name, player number, height, weight, and college.   + Report must be sorted by team, position, last name, and first name.   + Control breaks are required on both team and position.   + Total number of players per team must be displayed at the end of each team's data.   + All data must be displayed and formatted as shown in the sample report below.   + Validate all data entry allowing for upper & lower case; allow user to enter h for help which then displays a valid list of team codes and related team names (nicely formatted, each on a separate line). |

### Developer Notes

* 1. Accept user input for any number of teams – put their selections in a 1D list.
  2. Filter players by keeping only players that are on the selected teams list.
  3. Sort the filtered players by team, position, last name, first name – all ascending.
  4. Display the data for the sorted players and ensure alignment is completed.
  5. Data Validation - ensure the user enters at least one valid team code before proceeding. Remember they must be able to enter more than one team code. If the user enters an invalid code, notify them and prompt them if they want to see the list of valid codes - display the team codes & team names from the 2 parallel lists that were provided. Also, make sure all values (team codes, y/n, etc.) can be entered as upper or lower case.
  6. Code the control breaks – tip: keep track of the current & previous team (and position)
     + Add a 1st control break on *Team* - i.e. display the team name banner line, at the end of the team's data, display the total players line.
     + Add a 2nd control break on *Position* - i.e. display only the 1st of each position code

### Sample Report

|  |
| --- |
| ------------------------- Roster Report (by Team & Position) -------------------------  Enter a team code: **aaa**  >> Invalid team code. Would like to view a list of team codes & names (y/n)?: **n**  Enter a team code: **DLC**  >> Do you want to enter another team code (y/n)? **y**  Enter a team code: **hst**  >> Do you want to enter another team code (y/n)? **N**  >>>>>>>>>>>>>>>>>>>>> ROSTER FOR: DLC - Dallas Cowboys <<<<<<<<<<<<<<<<<<<<<  Pos Last, First (No.) Height Weight College  ---- ------------------------------ ------ ------ ------------------------  C Biadasz, Tyler (63) 6'4" 318 Wisconsin  Looney, Joe (73) 6'3" 310 Wake Forest  ---- ------------------------------ ------ ------ ------------------------  CB Brown, Anthony (30) 5'11" 188 Purdue  Diggs, Trevon (27) 6'1" 203 Alabama  Goodwin, C.J. (29) 6'3" 187 California (PA)  Lewis, Jourdan (26) 5'10" 186 Michigan  Robinson II, Reggie (41) 6'1" 200 Tulsa  Smith, Saivion (32) 6'1" 200 Alabama  ---- ------------------------------ ------ ------ ------------------------  ... there is more data; not shown here due to lack of space  Total players: 54  >>>>>>>>>>>>>>>>>>>>> ROSTER FOR: HST - Houston Texans <<<<<<<<<<<<<<<<<<<<<  Pos Last, First (No.) Height Weight College  ---- ------------------------------ ------ ------ ------------------------  C Mancz, Greg (65) 6'4" 302 Toledo  Martin, Nick (66) 6'4" 295 Notre Dame  ---- ------------------------------ ------ ------ ------------------------  CB Armstrong, Cornell (30) 6'0" 185 Southern Miss  Crossen, Keion (35) 5'10" 185 Western Carolina  Gaines, Phillip (29) 6'0" 193 Rice  ... there is more data; not shown here due to lack of space  Total players: 53  ...press enter to return to the MENU |

\*Note: "...press enter to return to the MENU" is coded in main() and is displayed once this menu option has completed executing.

## US2 – View Team Weight Analysis Report

### User Story

|  |
| --- |
| ***User Story:*** As a team physical trainer, I would like to compare weight data of our team to other teams as well as league averages so that I can address potential issues.  ***Acceptance Criteria:***   * + Must be able to enter any 2 teams to be analyzed: their team, and another team.   + Display team average weights regardless of position.   + Display average weights for the specified positions.   + Denote with an **asterisk** the higher of the 2 averages; a **dash** on both sides for ties. |

### Developer Notes

1. Accept user input for any two teams.
2. Work on the ALL output line. Each average is the total / count for a given team – so you will need running totals and counters for team1 and team2 weights.
3. Next, work on the weights by position lines.
   * At the top of *this function*, create a new 1D list called: rpt\_codes containing only the 5 position codes of interest.
   * Each average is still the total / count but this time for each specific team & position combination.
   * **You are required to place the data (counts & totals) in four 1D Lists (or you may use a single 2D List)** – see the next page for logic that you may find useful.
4. Data Validation - ensure that both team codes are valid before proceeding. If the user enters an invalid code, notify them and prompt them if they want to see the list of valid codes - display the team codes & team names from the 2 parallel lists that were provided. Also, make sure all values (team codes, y/n, etc.) can be entered as upper or lower case.

### Additional Processing Logic for the Stats Data

|  |  |
| --- | --- |
| This can be done using only 1D Lists as shown below…    FYI…The four stats-related lists can also be coded as one 2D list. | Assume the user is interested in an analysis of ***Weights*** and picked Washington (**WAS**) for Team 1 and Dallas (**DLC**) for Team 2 for the report.  Before the loop:   * Create all lists with values pre-initialized.   Loop through all players:   * Assume we read the 1st record and it is a CB that plays for Washington and he weights 182. * Look up his position code in the pos codes list to get the correct element # (i.e. 3). * Then using that element #, scoot over to the appropriate parallel list and update the appropriate two elements– i.e. bump up the count by one and add the player’s weight to the total column. * This logic continues in the loop as we read each player. Eventually, the lists will be full of values.   After the loop:   * Simply divide each total in the stats list by its related count to display each average. |

### Sample Report

|  |
| --- |
| ----------------------------- Team Weight Analysis Report -----------------------------  Enter Team 1: **DLC**  Enter Team 2: **HST**  >>>>>>>>>>> WEIGHT ANALYSIS FOR: Dallas Cowboys & Houston Texans <<<<<<<<<<<  DLC HST  -----------------  ALL \* 246 242  -----------------  QB \* 223 216  RB \* 218 217  WR \* 207 190  CB - 194 194 -  DT \* 314 301  ...press enter to return to the MENU |

\*Note: "...press enter to return to the MENU" is coded in main() and is displayed once this menu option has completed executing.

## US3 – View Filtered Players Report

### User Story

|  |
| --- |
| ***User Story:*** As a team executive, I need to view players across the league so that I can quickly find potential players to trade for.  ***Acceptance Criteria:***   * + Must be able to filter players by: team code, pos, last name, first name, and min & max weight   + Must have 2 sort options: a) by name; b) by team & position   + With the final report, redisplay the filter choices that were selected, the sort option selected, and the total number of players that met the criteria.   + For player meeting the criteria, display *all* player data. |

### Developer Notes

1. **Filter** the data. Filter choices should allow filtering on any or all of the following (i.e. all are not always required). Ensure at least one filter is always entered.
   * + Team Code – code an exact match
     + Position Code – code an exact match
     + Last Name – code a pattern match (starts with) – see Ch 8 Strings
     + First Name – code a pattern match (starts with) – see Ch 8 Strings
     + Weight Range (min) – minimum inclusive
     + Weight Range (max) – maximum inclusive
       - Note: must handle the user entering both min & max weights, just one, or none.
2. **Sort** the data. Provide the following sort options:
   * + Sort by Name 🡪 i.e. sort this by last name, first name – all ascending
     + Sort by Team, Position 🡪 i.e. sort this by team code, position code, last name, first name – all ascending
     + Also, allow the user the option to not sort the data in which case the data is presented as it appears in the file.
3. **Display** the report as shown in the sample report.

### Sample Report

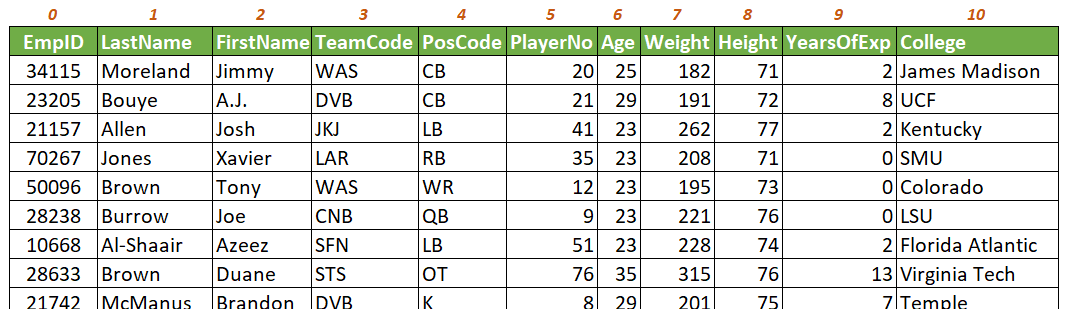
|  |
| --- |
| ------------------------------- Filtered Players Report -------------------------------  Enter any Filter Criteria...  Team Code: **dlc**  Position Code:  Last name:  First name:  Weight (min): **305**  Weight (max):  Sort Options:  1) Name  2) Team & Position  3) Weight  Enter a sort option (or enter for none): **3**  >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> FILTERED REPORT <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<  >> Filters: > Team: DLC > Weight (min): 305  >> Sort fields: Weight  >> Met criteria: 12 players  Emp ID Last Name First Name Team Code Pos Code Player No Age Weight Height Years College  ------ ----------------- -------------- --------- -------- --------- --- ------ ------ ----- ------------------------  44112 Erving Cameron DLC OT 75 28 305 78 6 Florida State  51469 Senat Greg DLC OT 64 26 305 78 3 Wagner  82900 Gallimore Neville DLC DT 96 23 307 74 0 Oklahoma  26686 Looney Joe DLC C 73 30 310 75 9 Wake Forest  10635 Woods Antwaun DLC DT 99 27 310 73 3 USC  89481 Hamilton Justin DLC DT 79 27 315 74 4 Louisiana-Lafayette  70111 McGovern Connor DLC G 66 23 315 77 2 Penn State  70970 Martin Zack DLC G 70 29 315 76 7 Notre Dame  33246 Biadasz Tyler DLC C 63 22 318 76 0 Wisconsin  91433 Williams Connor DLC G 52 23 320 77 3 Texas  87101 Steele Terence DLC OT 78 23 320 78 0 Texas Tech  78171 Ankou Eli DLC DT 26 325 75 4 UCLA  ...press enter to return to the MENU |

\*Note: "...press enter to return to the MENU" is coded in main() and is displayed once this menu option has completed executing.

# About the Data

* Most of the data is self-explanatory. Note: Height is stored in inches.

*Players File*



*Codes*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **TeamCode** | **Name** | | ATF | Atlanta Falcons | | AZC | Arizona Cardinals | | BFB | Buffalo Bills | | BLR | Baltimore Ravens | | CAP | Carolina Panthers | | CHB | Chicago Bears | | CLB | Cleveland Browns | | CNB | Cincinnati Bengals | | DLC | Dallas Cowboys | | DTL | Detroit Lions | | DVB | Denver Broncos | | GBP | Green Bay Packers | | HST | Houston Texans | | INC | Indianapolis Colts | | JKJ | Jacksonville Jaguars | | KCC | Kansas City Chiefs | | LAC | Los Angeles Chargers | | LAR | Los Angeles Rams | | LVR | Las Vegas Raiders | | MMD | Miami Dolphins | | MNV | Minnesota Vikings | | NEP | New England Patriots | | NOS | New Orleans Saints | | NYG | New York Giants | | NYJ | New York Jets | | PHE | Philadelphia Eagles | | PTS | Pittsburgh Steelers | | SFN | San Francisco 49ers | | STS | Seattle Seahawks | | TBB | Tampa Bay Buccaneers | | TNT | Tennessee Titans | | WAS | Washington Football Team | | |  |  |  | | --- | --- | --- | | **PosCode** | **Name** | **Team Type** | | C | Center | Offense | | CB | Cornerback | Defense | | DE | Defensive end | Defense | | DT | Defensive tackle | Defense | | FB | Fullback | Offense | | G | Guard | Offense | | K | Kicker | Special Teams | | LB | Linebacker | Defense | | LS | Long snapper | Special Teams | | NT | Nose tackle | Defense | | OT | Offensive tackle | Offense | | P | Punter | Special Teams | | QB | Quarterback | Offense | | RB | Running back | Offense | | S | Safety | Defense | | TE | Tight end | Offense | | WR | Wide receiver | Offense | |

# Participation and Peer Evaluations

An important aspect of this project is learning to work with others. As such, the expectation is that each student will be a member of a ***team (of 2 members)*** and participate actively and fully in all aspects of the current project assignment. The peer evaluation grade is worth 5% of the course grade.

*Students working alone:*

* Any student that prefers to complete *this assignment* alone (and not with a team) may do so but will **sacrifice the peer evaluation grade for *this deliverable* which is worth 5% of the overall course grade**. You must notify your professor of this decision ***prior*** to the assignment start date. Note: each subsequent deliverable will have the same requirement.

*Students working on a team:*

* If you decide to be on a team for this deliverable, you **must** ***actively* participate** on the project **as indicated by your peers** in order to receive a team assignment grade; non-active participation by a team member needs to be indicated on peer evaluations by evaluating the team member with 0’s and with supporting comments. Any team member that receives **majority 0’s on the peer evaluation will result in no grade for the project.** The expectation is that every team member does their fair share of the project; points will be deducted based on peer evaluations scores.
* All students must submit a *Peer Evaluation Form.* The form is due **the day after the due date for this deliverable.** An evaluation that is turned in late will be accepted, with penalty, up two calendar days past the specified due date. *The late penalty is a deduction of 10%* of the possible points for the evaluation grade, *per**day**late*. ***Failure to submit a completed evaluation form within that time period will result in a grade of 0 for your evaluation score*** as non-submittal impacts the peer evaluation score for all students on the team.

# Submitting the Assignment

Your team will submit a single application program (\*.py) containing all required use cases as well as the Help Document (\*.docx) on Canvas.