CS 2336 – PROJECT 2 – Mario Super Sluggers 2

Pseudocode Due: 10/2 by 11:59 PM (no late submissions)

Core Implementation Due: 10/9 by 11:59 PM (no late submissions)

Final Submission Due: 10/18 by 11:59 PM (late submission up to 24 hours)

KEY ITEMS: Key items are marked in red. Failure to include or complete key items will incur additional deductions as noted beside the item.

Submission and Grading:

- The pseudocode will be submitted in eLearning as a Word or PDF document and is not accepted late.
- All project source code will be submitted in Zybooks.
    - Projects submitted after the due date are subject to the late penalties described in the syllabus.
- **Type your name and netID in the comments at the top of all files submitted. (-5 points)**

Objective: Use object-oriented programming to implement and utilize a linked list data structure

Problem: Nintendo is developing a new Mario baseball game for Nintendo Switch. As an (unpaid) intern for Nintendo this semester, you have been given the task of developing a system to calculate player stats and determine the league leaders in several different baseball categories, such as hits, strikeouts and walks.

Pseudocode: Your pseudocode should describe the following items

- Identify the functions you plan to create for each class
    - You do not have to include pseudocode for basic items like constructors, accessors, mutators
- For each function, identify the following
    - Determine the parameters
    - Determine the return type
    - Detail the step-by-step logic that the function will perform
- A list of at least 10 test cases you will check during testing (other than the examples below)
    - Specific input is not necessary
    - Describe what you are testing
    - Examples:
        - Processing 3 leaders for first place
        - Sorting list in reverse alphabetical order

Zybooks Information:

- You will have multiple source files
    - `Main.java`
    - `LinkList.java`
    - `Node.java`
    - `Player.java`
- Core implementation has unlimited submissions
    - This will help you make sure the basic actions of your program are working properly in the environment

- Final submission is limited to 10 submissions
- White space will not be checked

Core Implementation:

- Read file
- Calculate batting average
- Calculate on-base percentage
- Count hits, walks, strikeouts, hit by pitch
- Display individual players' stats
- Not required
    - Sort batters alphabetically
    - League leaders
    - Multiple player records

This will project will extend the basic logic and retain most of the requirements of Project Zero. Differences from Project Zero will be written in blue.

Details:

- **Classes**
    - Use good programming practice for classes – proper variable access, mutators and accessors, proper constructors, etc.
    - Remember that classes exist to be used by other people. Just because you don't use it in the program doesn't mean it shouldn't be coded and available for others to use in theirs.
    - As with previous projects, you are open to design the classes as you see fit with the minimum requirements listed below
    - Both classes must be of your own design and implementation.
    - **Requirements**
        - Linked list class
            - Head pointer
            - Recursive print function
            - A sort function of your own design
        - Node class
            - Next pointer
            - Player object
            - **EXTRA CREDIT: Make Node generic instead of using a Player object within Node (+10 points)**
                - **If generic, no references to Player**
                - **Linked list will also need to be generic because the head pointer will need to be of a generic node type**
                - **No references to Player inside linked list class**
        - Player class
            - Design class to hold stats and name
                - Do not hold stats that are calculated from other stats
                - Use functions to calculate when needed

    - This prevents stale data
- Start the program by prompting the user for the input filename
- Nodes will be added to the end of the linked list
- Stats will be calculated for the following categories:
    - Batting Average (BA)
        - Batting average = hits / at-bats
    - On-base percentage (OB%)
        - On-base percentage = (hits + walks + hit by pitch) / plate appearances
    - Strikeouts (K)
    - Walks (BB)
    - Hit by Pitch (HBP)
    - Hits (H)
- Calculate all stats per person and record the highest value for each category
    - There may be ties for the leaders
    - If there is a tie, output all names for tied value
- All data must be held and manipulated in a linked list of your design (-20 points if not)

Input: All input will be read from a file. Each player's data will be listed on separate lines in the file and will follow the same format.

- Format: <name><space><batting record>
    - `Mario HOOKWSHHKOHPWWHO`
- The name will be a single word.
- The batting record will be a series of capital letters representing various results during a baseball game
    - H – hit
    - O – out
    - K – strikeout
    - W – walk
    - P – hit by pitch
    - S – sacrifice
- Walks, sacrifices and hit by pitches are not considered an at-bat
- Batting records may contain invalid characters.
    - If an invalid character is encountered, disregard it
    - Invalid characters are not counted as an at-bat
- Each line in the file will end in a newline (except the last line of the file which may or may not have a newline)
- The batting record for each person may not have the same number of results
- The input file may have multiple entries for the same person
    - Combine that data into one node for the player

Output:

- All output will be written to the console
- All floating-point values are displayed to 3 decimal places
- Display each player's data in the following order with a tab between each field

    - Player name
    - At-bats
    - Hits
    - Walks
    - Strikeouts
    - Hits by pitch
    - Sacrifices
    - Batting average
    - On-base percentage
    - Write a newline after the on-base percentage (there is no tab before the newline)
- Output the player data by name in alphabetical order (A to Z)
- After the player data table, display an additional newline and the `LEAGUE LEADERS` header
- Display the top 3 leaders for each category
- Because of ties all places may not be awarded.
    - For example, if there is a 3 way tie for first, there would not be a second or third place. (see output format below)
    - When displaying the leader list, separate each name with a comma and a space
- **League Leader Order**
    - Batting Average
    - On-Base Percentage
    - Hits
    - Walks
    - Strikeouts
    - Hit By Pitch
- **League Leader Output Format**
    - <CATEGORY><newline>
        - All caps
    - <value><tab><first leader list><newline>
    - <value><tab><second leader list><newline>
        - Second leader list is optional
        - No second place if first place has 3 or more ties
    - <value><tab><third leader list><newline>
        - Third leader list is optional
        - No third place if first or second place has a tie
    - <newline>