CS 2336 – PROJECT 5 – Super Mario World Series

Pseudocode Due: 4/16 by 11:59 PM
Project Due: 5/2 by 11:59 PM

This is the third of three maintenance projects. This will project will extend the basic logic and retain most of the requirements of projects 1 and 3. Differences from these projects will be written in blue.

Submission and Grading: All programs are to be submitted in eLearning. The pseudocode should be submitted as a Word or PDF document and is not accepted late. Projects submitted after the due date are subject to the late penalties described in the syllabus. Programs must compile and run in NetBeans 8.0.2. Each submitted program will be graded with the rubric provided in eLearning as well as a set of test cases. These test cases will be posted in eLearning after the due date. Each student is responsible for developing sample test cases to ensure the program works as expected. **Type your name and netID in the comments at the top of all files submitted.**

What to Submit: Save all project files in a directory named with the following format: <netID>_Project<number>. This directory should contain all files associated with the project including the project file itself. Zip up the directory into a standard Zip file. Please do not create a RAR or 7-Zip file, just a .zip file.

Objective: Utilize common features of a hash table in the design of an object-oriented program

Problem: Now that the Mushroom Kingdom League has ended and crowned a new champion, it is time to determine the league leaders in several different baseball categories. Being the only person in the Mushroom Kingdom that knows how to write a computer program, you have been asked by Princess Peach herself to write a program that will determine the league leaders.

Details:

- This program will calculate stats based on the game play-by-play
    - Each plate appearance will be given in a file
    - Analyze each plate appearance to determine the result (hit, out, strikeout, etc.)
    - Record the information for each player and determine leaders
- Hash tables will be used for quick lookup of results and players
    - After reading the play, a hash table will be used to determine the result (i.e. H, O, K, etc.)
    - For example, a play of 1B (single) is a hit (H) (which as you know is also an at-bat and a plate appearance)
    - The result is then recorded for the player
        - Use a hash table to find the player and update the stats
- Collisions will be possible (and likely to happen)
    - Use separate chaining or double hashing to handle collisions
- A file of all possible plate appearances will be provided in order to create the hash table for results
- **Classes**
    - Design a class suitable for holding player information
        - Be careful of including members that may become stale
    - 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
- 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)
- All input will be read from a file
- All output will be written to a file
- Calculate all stats per person and record the highest values for each category
    - There may be ties for the leaders
    - If there is a tie, output all names for tied value
- Any global variables used must be constant
- Use as few variables as possible

User Interface: There will be no user interface for this program since all input will be read from a file.

Input: All input will be read from a file. Each line in the file will represent a plate appearance and will follow the same format.

- The input file will be named *PlayByPlay.txt*
- Format: <H/A><space><player name><space><plate appearance>
    - `H Mario 6-3`
- The name will be a single word.
- The plate appearance will be a sequence of characters (a code) describing what happened
    - All plate appearance "codes" will be valid and will be present in the results file provided
- Errors are considered an at-bat
- Walks, sacrifices and hit by pitches are not considered an at-bat
- Each line in the file will end in a newline (except the last line of the file)
- The total number of plate appearances may be different for each player
- The input file will have multiple entries for the same person
    - Combine data for each player

Output: All output will be written to a file named *BoxScore.txt*. Use formatted output to display the player information in a table. Divide the player information into home and away teams. For each team, the player names should be in alphabetical order with his/her data on the same line in columns. Create a header for the table identifying each column. The columns should be ordered as follows:

- Player name
- At-bats

- Hits
- Walks
- Strikeouts
- Hits by pitch
- Sacrifices
- Batting average
- On-base percentage

After the player data table, print the top 3 leaders for each category (listed above in **Details**). 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.

League Leader Order

- Batting Average
- On-Base Percentage
- Hits
- Walks
- Strikeouts
- Hit By Pitch

League Leader Output Format

- <CATEGORY>
    - All caps
- <value><tab><first leader list>
- <value><tab><second leader list>
    - Second leader list is optional
    - No second place if first place has 3 or more ties
- <value><tab><third leader list>
    - Third leader list is optional
    - No third place if first or second place has a tie
- <blank line>