

## CS 2336 – PROJECT 4 – Super Mario World Series

**Pseudocode Due:** 4/13 by 11:59 PM

**Project Due:** 4/30 by 11:59 PM

**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 zyLabs.
  - Projects submitted after the due date are subject to the late penalties described in the syllabus.
- 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.
  - zyLabs will provide you with an opportunity to see how well your project works against the test cases. Although you cannot see the actual test cases, a description will be provided for each test case that should help you understand where your program fails.
- **Type your name and netID in the comments at the top of all files submitted. (- 5 points)**

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

**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 your main file
  - Be sure to explain how you plan to use the hash tables, including how you will retrieve and use the team data and leaders
- For each function, identify the following
  - Determine the parameters
  - Determine the return type
  - Detail the step-by-step logic that the function will perform

### Classes

- Design a class suitable for holding player information
  - Be careful of including members that may become stale
  - Name the file **Player.java**
- 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

## Details:

- Start the program by prompting the user for the input filename.
- This program will calculate stats based on the game play-by-play
  - Each possible plate appearance is provided in a file named **keyfile.txt**
    - This will be used for every test case
    - **Do not prompt for this file**
  - Each player's 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 (-15 points if not)
  - 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
  - You may utilize the built-in Java hashmap
  - **EXTRA CREDIT**
    - Instead of using the built-in Java hashmap, create your own hashmap class
    - Make the hashmap class generic (5 points)
    - Implement double hashing for collision handling (7 points)
    - Implement a rehashing function (8 points)
      - The size of the new hashmap will be the next highest prime number after doubling the old size
    - The filename for the HashMap class will be **GenericHashMap.java**
    - **If you use the built-in Java hashmap you must submit an empty file named GenericHashMap.java**
      - This is a restriction of the system in which a list of all files to be compiled must be provided in advance.
- Stats will be calculated for the following categories:
  - Batting Average (BA)
    - $\text{Batting average} = \text{hits} / \text{at-bats}$
  - On-base percentage (OB%)
    - $\text{On-base percentage} = (\text{hits} + \text{walks} + \text{hit by pitch}) / \text{plate appearances}$
  - Strikeouts (K)
  - Walks (BB)
  - Hit by Pitch (HBP)
  - Hits (H)
- 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

**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.

- Line 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 which may or may not have a newline)
- 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 the console
- Divide the player information into home and away teams.
- For each team, the player names will be in alphabetical order with his/her data on the same line
- Output format for the away team
  - AWAY
  - Player data – 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)
- After all away data has been displayed, display another newline
  - This will put a blank line between the Home and Away sections
- Display the Home team data with the same format as the Away team
  - Display HOME instead of AWAY
- 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.
  - If there are ties on both teams, list all away players first and then the home players
    - The names on each team will be in alphabetical order

### 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>