

CS 2336 – PROJECT 4 – Super Mario World Series

Pseudocode Due: 11/11 by 11:59 PM (no late submission)

Project Due: 12/2 by 11:59 PM (submit up to 24 hours late)

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: 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
- A list of at least **15** test cases you will check during testing

Zybooks Information:

- You will have multiple source files
 - `Main.java`
 - `GenericHashMap.java`
 - `Player.java`
- Final submission is limited to 20 submissions

Classes

- Use good programming practice for classes – proper variable access, mutators and accessors, proper constructors, data hiding, etc.
- 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 play-by-play 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 (+10 points)**
 - Instead of using the built-in Java hashmap, create your own hashmap class
 - Make the hashmap class generic
 - Implement double hashing for collision handling
 - Implement a rehashing function
 - The size of the new hashmap will be the next highest prime number after doubling the old size
 - The only constructor should be a default constructor that initializes the size to 11. It will grow organically through the rehashing function based on how full the hash table is.
 - **If you use the built-in Java hashmap you must submit an empty file named GenericHashMap.java**
- 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 may have multiple entries for the same person
 - If so, combine data for the 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
 - League leaders may be on the home or away team
- 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 the output for more information on ties
 - 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>