

CST8130 Lab 6 Hashing and Collision Resolution

Problem Description:

In this lab, we will write a program to handle insertions into a dynamically allocated array or ArrayList using a hash algorithm and a collision resolution algorithm. You can just write the code inside a method main to do the following;

1. Declare a dynamically allocated array or ArrayList object called dataItems of 100 String objects.
2. Write a **switch-statement** based menu loop which allows these options:
[**note:** Create a *displayMenu* method that's called in the loop.]
 1. Add a String to the array dataItems
 - Prompt user to enter a String
 - use a hash algorithm to calculate the index of where to store the string in the dataItems array
 - the algorithm should be the first letter of the string to be added (converted to an int) added to the second letter of the input string if there is one (converted to an int) – modulus 100 (so that you make sure the result is a number between 0 and 99.
 - if there is already a String in this index position – move to the next sequential element position in the array until you find an empty location (but not past 99! – in this case display error message – String cannot be added)
 2. Search for a String in the array dataItems as efficiently as possible– and display the index of where it is found or a message if it is not found
 3. Exit
3. Create Javadoc for all classes and methods created.
4. Add meaningful code comments that explain the processing of your program codes.

Submission Requirements

Submit a zip folder named as **Lab6_Fname_Lname** containing *.java source files* and *Javadoc [here](#)*.

Grading Scheme

critrion	points
Add String to Data Items array - works and gives a visual confirmation	2
Search for String in array - works with a visual of the location	2
Correctly formatted output for all test cases	2
Exception handling, One or two classes used, <u>displayMenu</u> method, Switch statement loop. <u>proper</u> naming conventions + correct submission procedure followed	2
Javadoc and meaningful program code comments	2

EXPECTED OUTPUT from the Hashing Program when run is:

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
 > 1
 Enter string to insert: a
 Table: {97=a}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
 > 2
 Enter string to find: a
 "a" found at index: 97
 Table: {97=a}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
 > 1
 Enter string to insert: b
 Table: {98=b, 97=a}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
 > 2
 Enter string to find: b
 "b" found at index: 98
 Table: {98=b, 97=a}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
 > 1
 Enter string to insert: aa
 Table: {98=b, 97=a, 94=aa}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- > 2

Enter string to find: aa

"aa" found at index: 94

Table: {98=b, 97=a, 94=aa}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- > 1

Enter string to insert: bb

Table: {98=b, 97=a, 96=bb, 94=aa}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- > 2

Enter string to find: bb

"bb" found at index: 96

Table: {98=b, 97=a, 96=bb, 94=aa}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- > 1

Enter string to insert: A

Table: {98=b, 97=a, 96=bb, 94=aa, 65=A}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- > 2

Enter string to find: A

"A" found at index: 65

Table: {98=b, 97=a, 96=bb, 94=aa, 65=A}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- > 1

Enter string to insert: B

Table: {98=b, 97=a, 96=bb, 94=aa, 66=B, 65=A}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- > 2

Enter string to find: B

"B" found at index: 66

Table: {98=b, 97=a, 96=bb, 94=aa, 66=B, 65=A}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- > 1

Enter string to insert: AA

Table: {98=b, 97=a, 96=bb, 94=aa, 66=B, 65=A, 30=AA}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- > 2

Enter string to find: AA

"AA" found at index: 30

Table: {98=b, 97=a, 96=bb, 94=aa, 66=B, 65=A, 30=AA}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- > 1

Enter string to insert: BB

Table: {98=b, 97=a, 96=bb, 94=aa, 66=B, 65=A, 32=BB, 30=AA}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- > 2

Enter string to find: BB

"BB" found at index: 32

Table: {98=b, 97=a, 96=bb, 94=aa, 66=B, 65=A, 32=BB, 30=AA}

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- > 4

Invalid menu option

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- > j

Invalid menu option

1. Add a String to the array dataItems
 2. Search for a String in the array
 3. Exit
- >3

Exiting...