

A template with a preliminary implementation is provided in BeachBoard (template.zip) under *Template* of the *Content Tab*. The template provides the classes and a simple menu to interact. Modify the menu to support the required functionalities. Read the *readme.txt* file for the documentation.

Note Only assignments that use the template will be graded.

LAB 3: HASH TABLES

Learning objectives: CLO 1, CLO 3, CLO 4, CLO 5

Use Python 3.8 or higher for the assignment:

1. Implement the ChainedHashTable including *add(k, v)*, *find(k)*, *remove(k)* and *size()* operations covered in class. The name of each function should be according to the lecture.

Learning objectives: CLO 1, CLO 3, CLO 5

Test your program:

- Remove one element from an empty ChainedHashTable
- Search in an empty ChainedHashTable: *find(2)* should return nil
- Add 3 elements: *add(1, "first")*, *add(2, "second")*, *add(3, "fourth")*
- Check that *size()* returns 3
- find one element, *find(3)* should return "fourth"
- Remove one element: *remove(3)* and check that *size()* returns 2.
- Find one element: *find(3)* should return nil
- Add 3 elements: *add(3, "third")*, *add(4, "fourth")*, *add(5, "fifth")*.
- Check that *size()* returns 5
- Find one element: *find(3)* should return "third"

2. Calculator:

In Lab 1 you create a method that validates if a *mathematical expression* is properly matched. For example, "a + (b*c + d) / (a-c)" is a matched expression, but "a + (b*c + d / (a-c)" is not. In this assignment a dictionary will be used to transform a valid expression that contains variables to an expression that contains constant whenever the variable has been set. For example for the previous valid expression, if we add the values *add(a, "1.3")*, *add(b, "2.1")* and *add(c, 2.2)*, then it should print "1.3 + (2.1* 2.2 + d) / (1.3- 2.2)". To simplify the implementation, assume that the expression is full parenthesized (every two operands are in parenthesis) and contains only variables and no spaces. Thus, the expression consists of "(", ")", operands and operators. For example, ((a * b) + (c * d)).

Add three menu options: One for assigning the value to the variable in dictionary, another to introduce the expression and another to print the expression with variables. Observe that the first option requires the input from the user for the variable and the value and the second for the expression.

Learning objectives: CLO 1

Hint 1: First validate that the mathematical expression is valid.

Hint 2: Separate the string in token that can be "(", ")", operands and operators. For every operand, check if the value is in the dictionary.

Test your program:

- Evaluate an empty string
 - Introduce the expression $((a * b) + (c * d))$ and print.
 - `add(a, "1.3")`, `add(b, "2.1")`, `add(c, 2.2)`, and `add(d, 3)`, and print $((a * b) + (c * d))$, It should print $((1.3 * 2.1) + (2.2 * 3))$.
3. Book Store System. This part of the assignment is similar to Lab 1, and 2 except that we use the HashTable as a helper index. In development time, use the file "booktest.txt" with few books. Once you think it is ready, use the main file "books.txt".

Learning objectives: CLO 1, CLO 3, CLO 4

- (a) Load the catalog "books.txt" in an instance *indexTitle* of your *ChainedHashTable*. For each *key*, *title*, *category*, *rank* in "books.txt", create an instance *b* of Book and add it to *indexTitle*, i.e., *indexTitle.add(title, b)*.
- (b) Search books by title: Given a title *t*, find the book in *indexTitle*, i.e. *indexTitle.find(t)*; and if it is not nil, add it to the *shoppingCart*,

Test your program:

- Searching for books by:
 - (a) Empty prefix.
 - (b) "Tears of the Sun"
 - (c) "World of Piano"
4. What is the advantage and disadvantage of Hash Tables over Array-Based and Linked list data structures.

Learning objectives: CLO 4

5. Bonus points (3 points). Handle duplicated values. Observe that hash tables do not accept duplicated values. Design and implement a HashMap that accepts duplicate value. In the search and add a book title check if the title is unique, add it to *shoppingCart*, if it has more than one item, let the user choose the right value.

Learning objectives: CLO 3, CLO 5

Hint: Consider using a list as a value.

Test your program:

- Searching for books by:
 - (a) Empty prefix.
 - (b) Unique item: Worldly Saints.
 - (c) Duplicate item: "World War I Films of the Silent Era"

Submit all the source code (Python files (.py)) in a zip file. The name of the zip file with the source code must be your first name, second name, and the data structure separated by a hyphen. For example, oscar-ponce-hashtable.zip.

Submissions that do not follow the previous specification will be rejected and you will have 0 in the lab.

RUBRICS

	Level 4 2 Pt	Level 3 1.5 Pt	Level 2 1 Pt	Level 1 0.5 Pt
ChainedHashTable implementation	It is always correct without crashes	Eventually it crashes or return incorrect results	It frequently crashes and/or return incorrect results	It is not correct or incomplete
Calculator	It is always correct without crashes	Eventually it crashes or return incorrect results	It frequently crashes and/or return incorrect results	It is not correct or incomplete
Searching boook by title	It is always correct without crashes	Eventually it crashes or return incorrect results	It frequently crashes and/or return incorrect results	It is not correct or incomplete
Answer to Question 4	N/A	N/A	Correct	Incorrect
Searching books by title handling duplications	It is always correct without crashes	Eventually it crashes or return incorrect results	It frequently crashes and/or return incorrect results	It is not correct or incomplete