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 4: BINARY TREES

Learning objectives: CLO 1, CLO 3, CLO 4, CLO 5
Use Python 3.8 or higher for the assignment:

1. Implement the BinaryTree (including the recursive and no recursive operations: $depth(u)$, $size()$, $height()$ as well as traversing the tree $in\_order()$, $pre\_order()$, $post\_order()$ and $bf\_traversal()$) and BinarySearchTree that inheritance from BinaryTree ($find\_eq(x)$, $find(x)$, $add\_child(p, u)$, $find\_last(x)$, $add(x, v)$, $splice(u)$, $remove\_node(u)$ and $remove(x)$) covered in class.

   Learning objectives: CLO 1, CLO 3

   Test your program:

   - Remove one element from an empty BinarySearchTree
   - Search in an empty BinarySearchTree: $find(2)$ should return nil
   - Add 3 elements: $add(1, \text{“}first\text{”})$, $add(2, \text{“}second\text{”})$, $add(3, \text{“}fourth\text{”})$
   - Check that $size()$ returns 3
   - find one element, $find(2.5)$ 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: $add(3, \text{“}third\text{”})$, $add(4, \text{“}fourth\text{”})$, $add(5, \text{“}fifth\text{”})$.
   - Check that $size()$ returns 5
   - Find one element: $find\_eq(3.4)$ should return nil
   - Find one element: $find(3.4)$ should return $"fourth$”
   - Add menu options to display the values of the nodes using the in_order, pre_order and $post_order$.
   - Add menu options to display the values of the nodes using $bf\_traversal()$.
   - Add menu options to print the height of the tree.

2. Calculator: Implement the function $build\_parse\_tree(exp)$ that accepts a fully parenthesized mathematical expression and creates a parse tree and $evaluate(tree)$ as covered in class.

   Use the menu options from previous labs to assign the value to each variables in the dictionary, introduce the mathematical expression and evaluate the it with the variable.

   Learning objectives: CLO 1

   Hint: First validate that the mathematical expression is valid.

   Test your program:

- Evaluate an empty string
- Introduce the expression $((a * b) + (c * d))$ and evaluate. It should return 0 (handle every crash)
- $add(a, "1.3"), add(b, "2.1")$, $add(c, 2.2)$, and $add(d, 3)$, and print $((a * b) + (c * d))$, It should evaluate to $((1.3 * 2.1) + (2.2 * 3)) = 9.33$.

3. Book Store System. This part of the assignment is similar to Lab 3 except that we use the BinarySearchTree 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 $indexSortedTitle$ of your $BinarySearchTree$. For each $key, title, category, rank$ in "books.txt", create an instance b of Book and add it to $indexSortedTitle$, i.e., $indexSortedTitle.add(title, b)$.

   (b) Search by prefix: Given a prefix $prefix$, find the book in $indexSortedTitle$, i.e. $indexSortedTitle.find(prefix)$; 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 S"
     (c) "World of P"

4. What is the advantage and disadvantage of BinarySearchTree over HashTable data structures.

   Learning objectives: CLO 4

5. Add menu options to display the values of the nodes using the in_order, pre_order and $post_order$.

6. Add menu options to display the values of the nodes using $bf\_traversal()$.

7. Add menu options to print the height of the tree.

8. Bonus points (3 points). Handle duplicated values. Observe that BinarySearchTrees do not accept duplicated values. Design and implement a BinarySearchTrees 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-binarytree.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
BinaryTree and BinarySearchTree 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 books by prefix	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 book 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