# CS 2401 – Elementary data structures and algorithms Fall 2024

# Lab 8

Due Date: November 10, Sunday, 11:59PM.

Objective:

The goal of this assignment is to practice construction and traversal of binary trees in tandem with recursion. You will implement a Treenode and BinTree class that will be called from the main class (Lab8\_YOURLASTNAME) to construct a binary tree. Then you will output the contents of the tree using the three traversal methods you were shown in lecture. Finally, you will implement two methods that calculate the contents of the tree and search for a key value in the tree respectively.

Assignment – Timeline:

Milestone 1 – Nov 4 – Nov 6

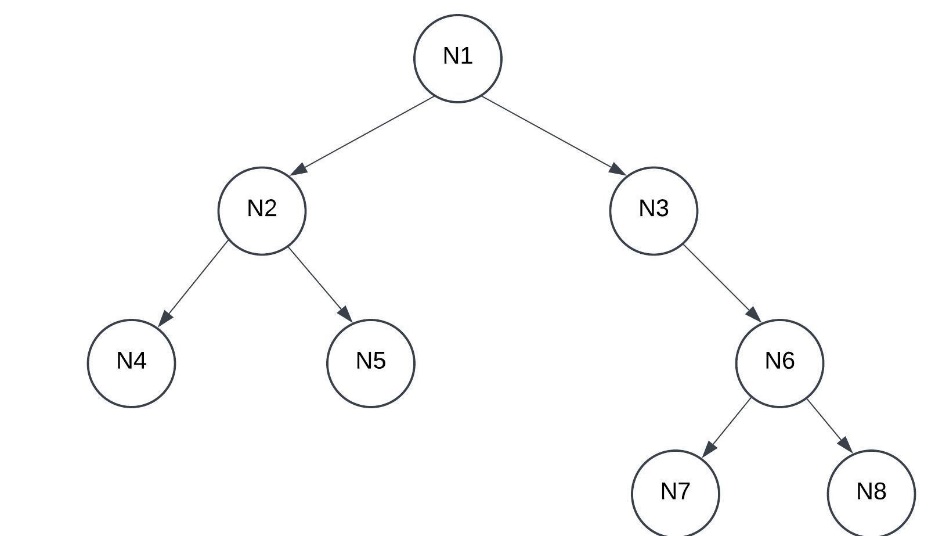
Task 1 – Implement the Treenode and BinTree classes (contents specified below)

* In the Treenode class:
  + Fill out the attributes and at least one constructor for the class.
  + The attributes should include data of type int and pointers for the left and right child nodes.
  + Have at least one constructor that takes int data as a parameter and sets the child pointers to null.
* In the BinTree class:
  + Provide one attribute for the root and at least one constructor.
  + Have at least one constructor that takes a node as a parameter and sets it to the root.
  + Implement the PrintPre, PrintIn and PrintPost methods that will print the contents of the tree in preorder, inorder and postorder respectively.
  + **DO NOT** implement the search or sumNodes methods yet, this will be done during your final task.

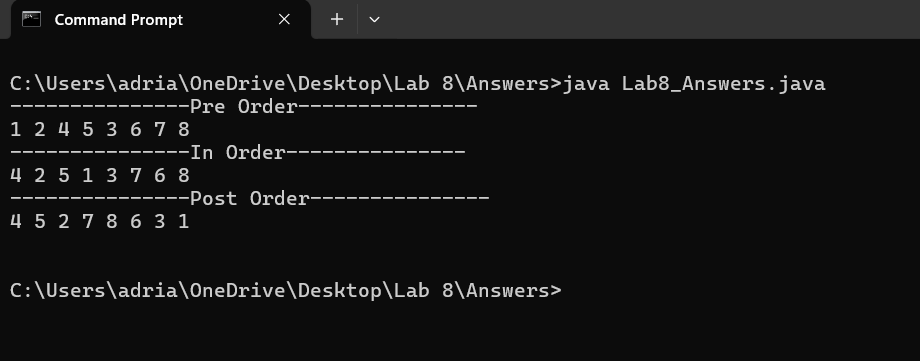
Milestone 2 – Nov 7 – Nov 8

Task 2 – Construct a Binary Tree and Print the Contents

* In the main method of the Lab8\_YOURLASTNAME class, construct a tree following the sample structure provided below. For the purposes of this exercise, you are asked to fill the data of each node in keeping with the number in the name of the node (i.e. N1’s data will be 1, N2’s data will be 2, etc). This is done for the sake of matching the sample outputs provided, however you are encouraged to also use your own values and rerun the program to trace the outputs of the various traversal methods.



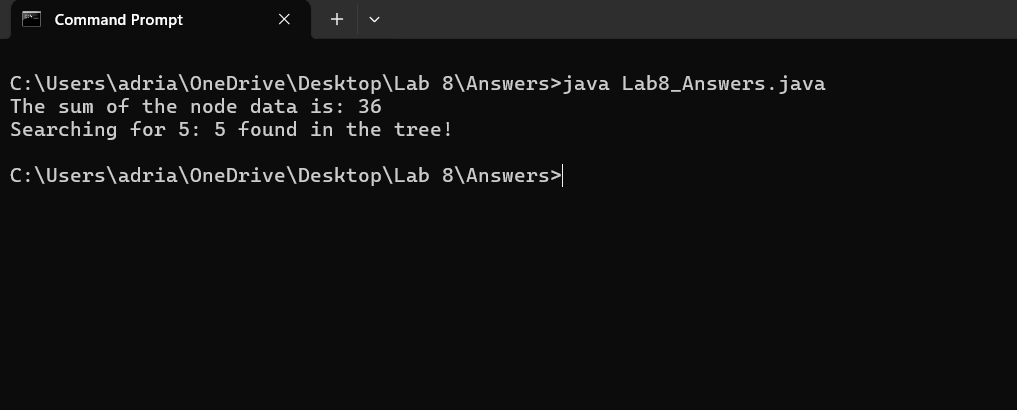
* After you have constructed your tree, uncomment the three lines under the “CALL THE TRAVERSAL METHODS” section and run the main.
  + Be sure to change the <YOURTREENAME> variable names to the name of your tree object.
* Test your code using the recommended data values for each node and compare with the screenshot below.

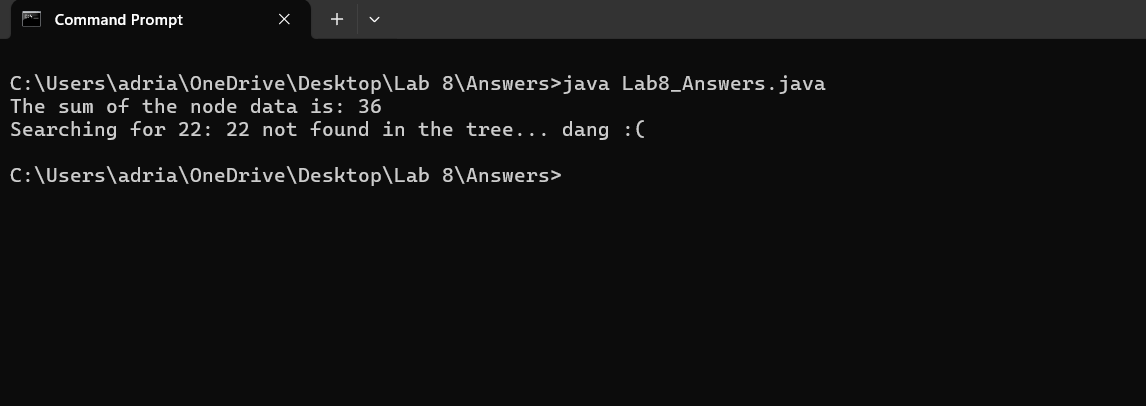


FULL SUBMISSION – Nov 9 – Nov 10

Task 3 – Implement the search and sumNodes methods in your BinTree Class

* In the BinTree class you will now implement the search and sumNodes methods. Their method signatures have been provided for you. Please note, using different names in the method implementation will require you to change the names in the method signatures as well.
* These methods will be implemented **RECURSIVELY**. The method signatures already have the number and type of parameter variables you will need to complete this task. If you do not implement these methods recursively your program will be treated as having incorrect logic (and you may find iterative implementation to be more difficult as well).
* After you have completed the methods’ implementation, uncomment the “CALL THE SUM AND SEARCH METHODS” section.
  + Be sure to change the <YOURTREENAME> variable names to the name of your tree object.
* Test your code using the recommended data values for each node and compare with the screenshots below. The screenshots show the output for keys found and not found respectively.





|  |  |
| --- | --- |
| Deliverables: |  |
|  | |

You are expected to submit one file in Blackboard (in your lab section).

Please use PULSE to develop your solution, then either:

* + Press Submit in PULSE, OR
  + download the .java files (from the Folder), rename the main class as shown below, and submit the implemented files in Blackboard. NOTE: This will also be your class name in java

Lab8\_Lastname.java --- the java file of your program.

Grading Criteria:

* + - [10 points] The program is indented correctly.
    - [10 points] The program is documented properly.
    - [10 points] The program uses correct variable types and meaningful variable names.
    - [10 points] Each milestone submission is met with code that demonstrates effort to complete the tasks within them.
    - [20 points] Program compiles and runs.
  + [20 points] The program has correct logic and generates correct output.
  + [20 points] Demo completed with a member of the instructional team.

\*\* Note that in case of academic dishonesty the grade for this lab will be a 0 and disciplinary actions will be taken \*\*

 Late submission: [-10] points for every 24 hours after the deadline.

If you need any clarification, please ask your TA for further details.