

CS 2401 Assignment #10

(Prepared by: Dr. Monika Akbar. This document is not for public distribution.)

CS 2401 – Elementary data structures and algorithms

Fall 2022: Lab 10

Due Date: Wednesday, November 23, 2022 – end of the day.

A maximum of three additional days are allowed for late submission with a penalty of 3% for each of those additional days.

Objective: The goal of this assignment is to practice implementing binary tree and binary search tree.

Assignment: There are three files you will need to complete this lab: `BTNode.java`, `BinaryTree.java`, and `Lab10_Runner.java`. The `BTNode.java` can be used as the node of any binary tree. Here is that file:

```
class BTNode{
    Object data;
    BTNode left;
    BTNode right;

    BTNode() {}
    BTNode(Object obj){
        data = obj;
    }
}
```

`Lab10_Runner.java` is the class that contains the main method and tests the binary search tree operations provided in `BinaryTree.java`. The code for `Lab10_Runner.java` is given next:

```
import java.util.Arrays;

class Lab10_Runner{
    public static void main(String[] args){
        System.out.println("-----");
        BinaryTree bst = new BinaryTree();

        bst.insertBST("Pluto");
        bst.insertBST("Mercury");
        bst.insertBST("Jupiter");
        bst.insertBST("Neptune");
        bst.insertBST("Saturn");
        bst.insertBST("Venus");
        bst.insertBST("Earth");
        bst.insertBST("Mars");
        bst.insertBST("Uranus");

        System.out.println("Printing BST:");
        bst.printBT();
        System.out.println("-----");
    }
}
```

To Chegg: Please do not provide solution if this document is uploaded. Please remove this document from Chegg, if uploaded.

CS 2401 Assignment #10

(Prepared by: Dr. Monika Akbar. This document is not for public distribution.)

```
System.out.println("Inserting Pluto in BST:");
System.out.println(""+bst.insertBST("Pluto"));
System.out.println("-----");

System.out.print("Total number of nodes in the BST: ");
System.out.println(bst.size());
System.out.println("-----");

System.out.print("Searching for Mars in the BST: \n");
System.out.println(""+bst.searchBT("Mars"));
System.out.println("-----");

System.out.print("Searching for Phobes in the BST: \n");
System.out.println(""+bst.searchBT("Phobes"));
System.out.println("-----");

System.out.println("Printing BST (ascending order): ");
bst.printAscending();
System.out.println("-----");

System.out.println("Printing BST (descending order): ");
bst.printDescending();
System.out.println("-----");

System.out.println("Retrieving the content of the tree from a String array:
");
String[] strArray = bst.getAsArray();
System.out.println( Arrays.toString(strArray));

System.out.println("-----");
}
}
```

The only file you have to change is the **BinaryTree.java**. This file contains a few incomplete methods. The documentation of the method and the sample output will give you all the information about what each of these methods should do. In case of any questions, please ask the TA or the instructor.

```
class BinaryTree{
    BTNode root; // The binary tree root
    int count; // Number of elements currently in the node

    BinaryTree(){}
    BinaryTree(String str){
        root = new BTNode(str);
    }

    /**
     * @return Number of elements in the binary
     * search tree.
     */
}
```

To Chegg: Please do not provide solution if this document is uploaded. Please remove this document from Chegg, if uploaded.

CS 2401 Assignment #10

(Prepared by: Dr. Monika Akbar. This document is not for public distribution.)

```
public int size(){
    // Change the body of this method
    return 0;
}

/**
 * Insert the string in the parameter into
 * the Binary Search Tree.
 * @param str
 * @return true if insertion is successful.
 */
public boolean insertBST(String str){
    // Change the body of this method
    return false;
}

/**
 * Return an array of strings containing the
 * string content elements of the tree.
 * Order of the strings in the array does not matter.
 * @return a String array
 */
public String[] getAsArray(){
    // Change the body of this method
    return null;
}

/**
 * Print the binary tree in the format
 * shown in the output.
 */
public void printBT(){
    // Change the body of this method
    System.out.println("This printBT method does not work.");
}

/**
 * Search the binary tree for the given string.
```

To Chegg: Please do not provide solution if this document is uploaded. Please remove this document from Chegg, if uploaded.

CS 2401 Assignment #10

(Prepared by: Dr. Monika Akbar. This document is not for public distribution.)

```
* @param str
* @return true if str is in the binary search tree.
*/

public boolean searchBT(String str){
    // Change the body of this method
    System.out.println("This searchBT method does not work.");
}

/**
 * Print the elements of the binary
 * tree in ascending order.
 */

public void printAscending(){
    // Change the body of this method
    System.out.println("This printAscending method does not work.");
}

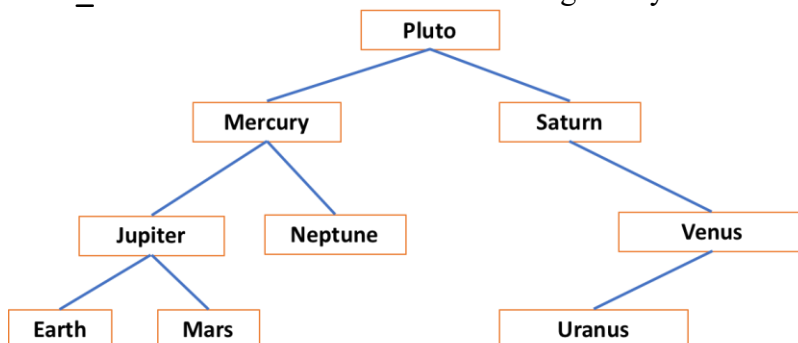
/**
 * Print the elements of the binary search
 * tree in descending order.
 */

public void printDescending(){
    // Change the body of this method
    System.out.println("This printDescending method does not work.");
}

}

}
```

Lab10_Runner class constructs the following binary search tree.



To Chegg: Please do not provide solution if this document is uploaded. Please remove this document from Chegg, if uploaded.

CS 2401 Assignment #10

(Prepared by: Dr. Monika Akbar. This document is not for public distribution.)

NOTE: Some requirements:

1. You must **NOT** change the `Lab10_Runner.java` and the `BTNode.java` files.
2. You **MUST NOT** create any new data/status/global variable in the `BinaryTree.java` file.
3. You **MUST NOT** change any of the headers that are already there in the `BinaryTree.java` file. **NO CHANGES IN THE PARAMETERS ARE ALLOWED.**
4. You **MUST NOT** call a public method from another public method unless the call from a method is to itself.
5. You **MAY** write private methods in the `BinaryTree.java` file. You may call a private method from a public method.

Sample output

Printing BST:

```
---Pluto
---Mercury
---Jupiter
---Earth
-
-
---Mars
-
-
---Neptune
-
-
---Saturn
-
---Venus
---Uranus
-
-
-
```

Inserting Pluto in BST:
false

Total number of nodes in the BST: 9

Searching for Mars in the BST:
true

Searching for Phobes in the BST:
false

Printing BST (ascending order):
Earth
Jupiter
Mars

To Chegg: Please do not provide solution if this document is uploaded. Please remove this document from Chegg, if uploaded.

CS 2401 Assignment #10

(Prepared by: Dr. Monika Akbar. This document is not for public distribution.)

Mercury
Neptune
Pluto
Saturn
Uranus
Venus

Printing BST (descending order):

Venus
Uranus
Saturn
Pluto
Neptune
Mercury
Mars
Jupiter
Earth

Retrieving the content of the tree from a String array:

[Pluto, Mercury, Jupiter, Earth, Mars, Neptune, Saturn, Venus, Uranus]

Deliverables:

Please submit the **BinaryTree.java** file in Blackboard.

Grading Criteria:

The guideline for grading this assignment is based on correctness of the functionality of the following methods.

- size – 5 points
- insertBST – 20 points
- printBT – 25 points
- printAscending – 10 points
- printDescending – 10 points
- searchBT – 15 points
- getAsArray – 20 points

Total points possible: 105 (out of 100). That is, the assignment has 5 bonus points.

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

To Chegg: Please do not provide solution if this document is uploaded. Please remove this document from Chegg, if uploaded.