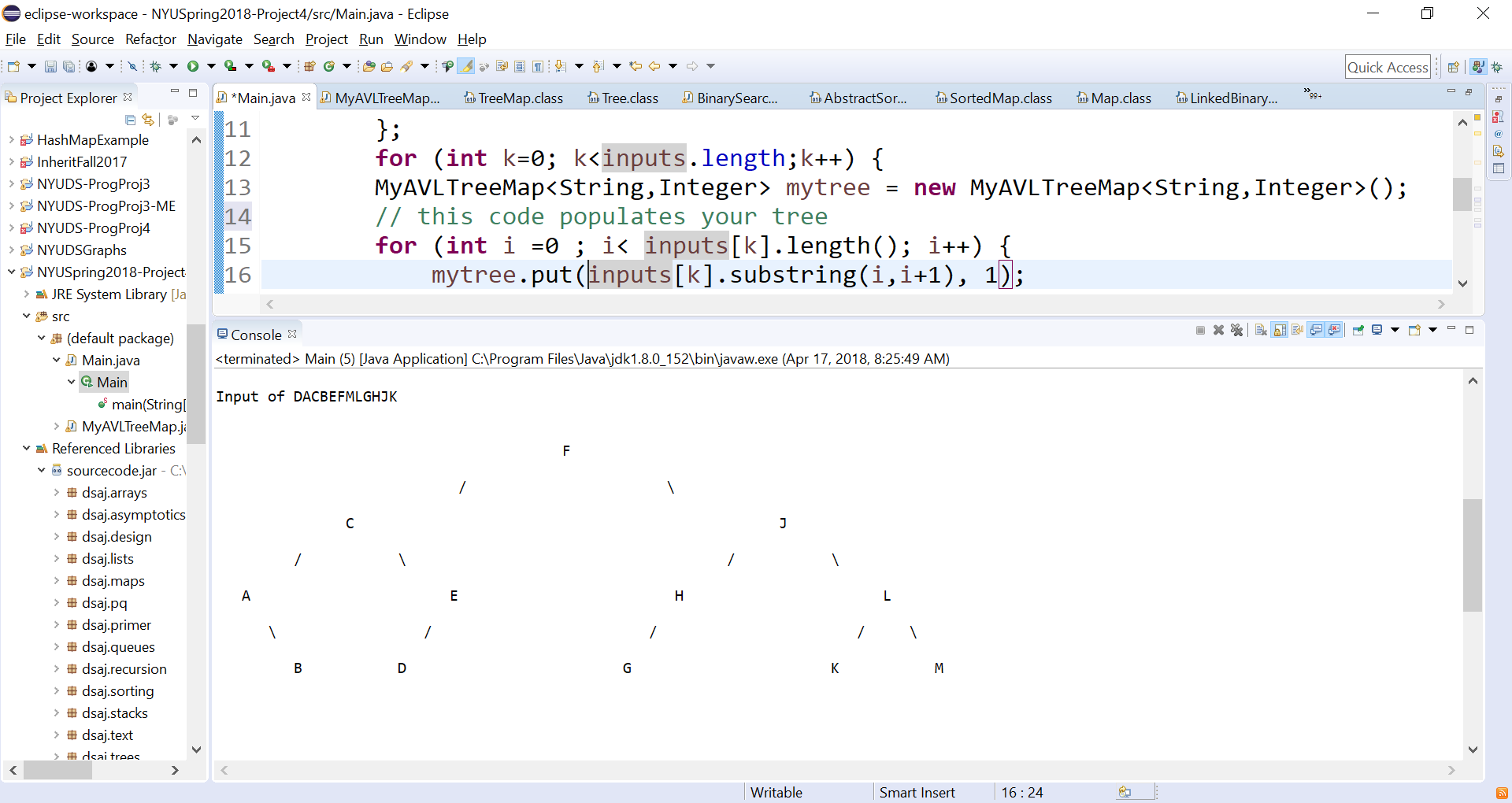
Project 3: AVL Tree Printing

Due date: Sunday July 23, 11:00 PM EST.

### You may discuss any of the assignments with your classmates and tutors (or anyone else) but all work for all assignments must be entirely your own . Any sharing or copying of assignments will be considered cheating (this includes posting of partial or complete solutions on any public forum). If you get significant help from anyone, you should acknowledge it in your submission (and your grade will be proportional to the part that you completed on your own). You are responsible for every line in your program: you need to know what it does and why. You should not use any data structures and features of Java that have not been covered in class (or the prerequisite class). If you have doubts whether or not you are allowed to use certain structures, just ask your instructor.

In this project you will populate a Binary Search Tree (BST), specifically an AVL tree and print out its contents. For example, a tree built from the input: will produce an output similar to the tree below:



The goal of the exercise to produce a reasonable facsimile of the tree. Since we are limited by output to the console line, the actual display can not be a perfect graphical representation of the tree. The lines of the trees are not exact but give the symbolic relationships.

# Objectives

The goal of this programming project is for you to master (or at least get practice on) the following tasks:

* understanding recursion and / or iterative calls within an AVL
* debugging and checking results
* Analyzing a problem to determine the best approach
* working with existing code

**Start early!** This project may not seem like much coding, but debugging always takes time. Analyze and plan now so questions are not being asked a day before the due date.

# Helpful code:

# A Main Class and MyAVLTreeMap class have been provided. The main class will use the class provided by the book to build the AVL tree. The example provided builds the 3 trees for you, ( though you are requested to include 2 more trees of your own.) that will call methods the AVL class provided. Each element of the tree has a key which is a letter of the input, and value of 1. The value of each element is this project is not relevant.

# The main class (ProgProject4.java) defines the inputs, creates the AVL and calls the print method (which you will write), in order to display the tree. You should use this code “as is”, and put your work into the printTree method of the MyAVLMapTree class.

# Your goal is to complete the PrintTree method in the MyAVLTreeMap class.

## One Approach

Your solution should be able to print all the nodes of the tree, without any collisions (nodes printing over other nodes), and display each node on its appropriate level. Some arrows (\ or /) should be included to help read your tree.

You are welcome to use any approach, however, I provide some key points my approach below to assist you. My PrintTree and PrintTree1(the recursive method) was in total 20 lines of code.

**Print Tree:**

Set up a 2 dimensional array , 100 x 100 , should be sufficient, to hold the node keys.  
 call PrintTree1 with starting height, row, column of where to place the root key, and root’s position

Print the array

**Print Tree1**

Store the key of the current position in the array  
 If current position has a left child , call PrintTree1 with parameters relating to the left child  
 If current position has a right child , call PrintTree1 with parameters relating to the right child

I viewed the output as a 2D array where every cell was one character to print out.

## Expected Results:

## 

## 

## 

## Running the program

The zipped java project file, which contains all your source code in your Eclipse project. This code should include all the code in the attached file, with the MyAVLTreeMap class having the additional printTree method and any other supporting methods. Provide the zip file of the entire Java project provided to you. If I need to reconstruct your project because you only provided source files, you will lose 20% of the project points. If this is not clear to you , see me before or after class.

Since you will be using the ProgProject4 class provided, running your program should be produce a series of tree input statement and their representation using the printTree method.

## Working on This Assignment

You should start right away! My effort was about 2 hours, and the total lines of code approximately 50. Your solution may be longer, or perhaps shorter. You should plan on spending at least 4-8 hours on the project.

**Grading Criteria – 20 points**

1. (5) Program provide some display of all the nodes
2. (5) Nodes are displayed in the proper order and level
3. (5) Results are totally accurate. There are not overlaps of nodes
4. (3) Original test cases have been included.
5. (2) Display is compact (no large amounts of unnecessary white space).