







Binary Tree with Inner Node Use Instructions

Precautionary Steps

1. Make sure you have **Java 17** installed on your system. To check, open your **Command Prompt/Terminal** and type in the command “**javac -version**”.
 - a. **If Java is not installed, your computer will tell you to install it. Click on one of the links below for installation:**
 - i. Windows:  [How To Download & Install Java for Minecraft \(Java 17\)](#)
 - ii. Mac:  [How To Download & Install Java 17 on Mac](#)
 - iii. Linux:  [How to install Oracle JDK 17 \(Java 17\) on Ubuntu 22.04 ...](#)
 - b. If your version of Java is different, the Gym Helper should still work, but having Java 17 will ensure that it does.
2. Make sure you have an **IDE** to run this project. (**Eclipse** is recommended)
 - a. **Eclipse Installation instructions:**
 - i. Windows:  [How to Download and Install Eclipse](#)
 - ii. Mac:  [How to Install Eclipse IDE on Mac | Install Eclipse IDE o...](#)
 - iii.  [How to Install Eclipse IDE on Ubuntu 24.04 LTS Linux \(...\)](#)

Steps for Use

1. Download the Binary Node zip file by [clicking here](#)
2. Go to your recent downloads (Or wherever the zip file is stored) and unzip the file. Now, you should have a folder containing all the necessary files for the project
 - a. **On Mac, you can do this by simply double clicking the file**
 - b. **On Windows 11, right click the zip file, then click *extract*. After that, you can save the folder in your desired location. Finally, hit *extract* in the bottom right hand corner**
3. If you get an error in the **BinaryTreeTester** file, **import junit4** into your project. **The following pictures explain how to import this on Eclipse.**

```

import static org.junit.Assert.*;

import java.util.Queue;
import java.util.Set;

import org.junit.Test;

public class BinaryTreeTester {

    @Test
    public void testNodeCount() {
        BinaryTree bt = new BinaryTree();
        assertEquals(0, bt.nodeCount());

        for (int i = 1; i <= 100; i++) {
            bt.addRandomLocation(i);
            assertEquals(i, bt.nodeCount());
        }
    }

    @Test
    public void testSum() {
        BinaryTree bt = new BinaryTree();
        assertEquals(0, bt.sum());
        int sum = 0;

        for (int i = 1; i <= 100; i++) {
            bt.addRandomLocation(i);
            sum += i;
            assertEquals(sum, bt.sum());
        }
    }

    @Test
    public void testMax() {
        BinaryTree bt = new BinaryTree();

        for (int i = 1; i <= 100; i++) {
            bt.addRandomLocation(i);
            assertEquals(i, bt.max());
        }

        bt = new BinaryTree();
        for (int i = 1; i <= 100; i++) {

```

There will be a red error line under **@Test**. Hover over it.

@Test

Test cannot be resolved to a type

4 quick fixes available:

- [Add JUnit 4 library to the build path](#)
- [Add JUnit 5 library to the build path](#)
- [Create annotation 'Test'](#)
- [Fix project setup...](#)

public void testSum() {

Click on **Add JUnit 4 library to the build path**.

```

import static org.junit.Assert.*;

import java.util.Queue;
import java.util.Set;

import org.junit.Test;

public class BinaryTreeTester {

    @Test
    public void testNodeCount() {
        BinaryTree bt = new BinaryTree();
        assertEquals(0, bt.nodeCount());

        for (int i = 1; i <= 100; i++) {
            bt.addRandomLocation(i);
            assertEquals(i, bt.nodeCount());
        }
    }

    @Test
    public void testSum() {
        BinaryTree bt = new BinaryTree();
        assertEquals(0, bt.sum());
        int sum = 0;

        for (int i = 1; i <= 100; i++) {
            bt.addRandomLocation(i);
            sum += i;
            assertEquals(sum, bt.sum());
        }
    }

    @Test
    public void testMax() {
        BinaryTree bt = new BinaryTree();

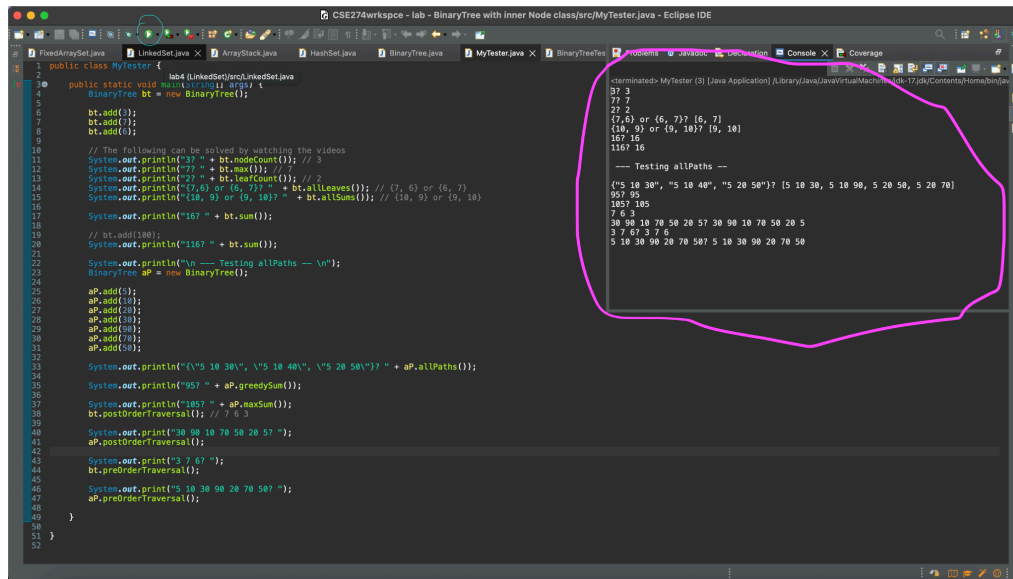
        for (int i = 1; i <= 100; i++) {
            bt.addRandomLocation(i);
            assertEquals(i, bt.max());
        }

        bt = new BinaryTree();
        for (int i = 1; i <= 100; i++) {

```

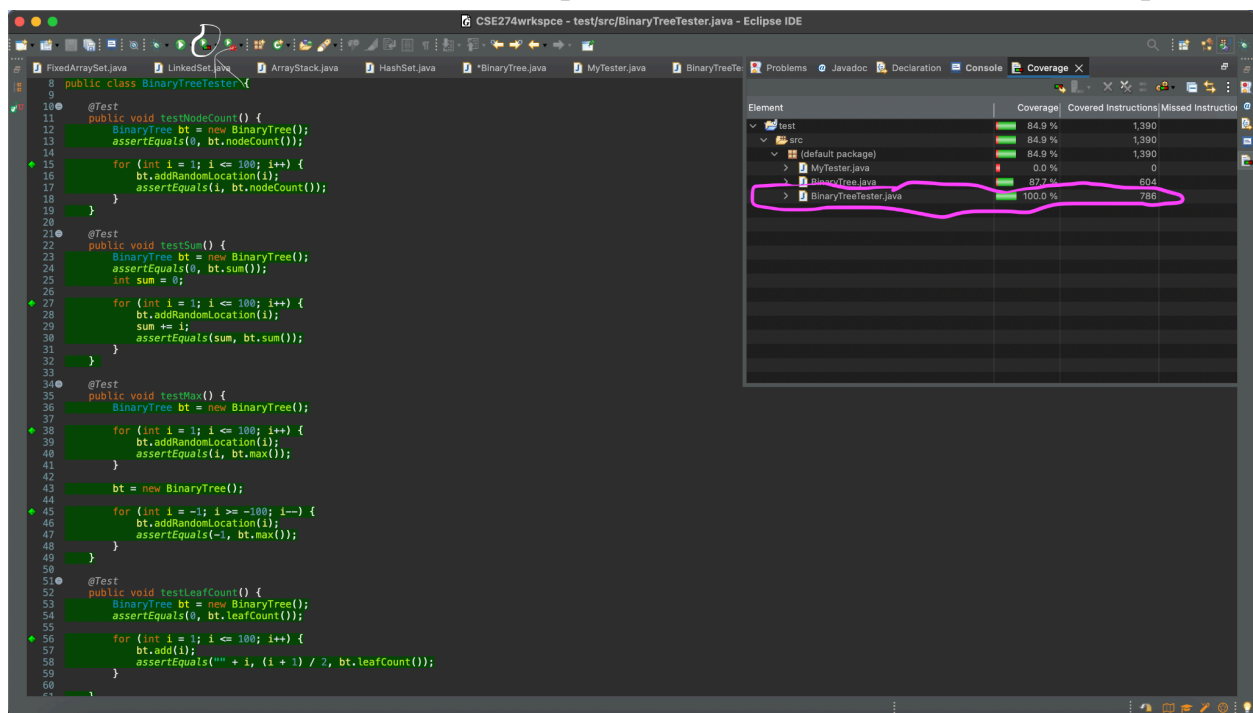
Now, the file should be error free.

4. Now, you can **run** the code through the **MyTester.java** file. This file has manual tests of each method that I wrote, so you can edit these if you'd like. (Screenshot below will show how this will look on Eclipse)



The **cyan** circle shows the **run** button that should be pressed. The **pink** circle shows what the code will look like while running.

- To run the **jUnit** test for the code's accuracy, click on the **BinaryTreeTester** file. Then, click run. There's a picture below on how to do this in Eclipse.



The **white** circle shows the **jUnit** test button to click. The **pink** circle shows that the tester file passed all of the **jUnit** tests.

Bonus: To see the report of the methods written, look through the code comments in the BinaryTree.java file