CSCI 1933 Lab 13
JUnit Test Guide

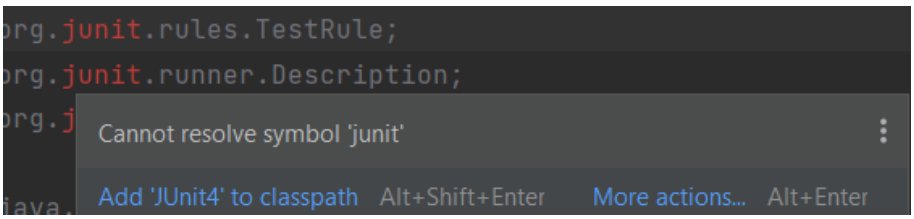
1 Introduction

This document is intended to give you the information necessary to run the provided for your project. It is not intended to be a comprehensive guide on JUnit or unit tests in general. To learn more, please visit the JUnit website. The provided test files are what TAs will be using to grade projects, and the tests will contribute a significant portion of the grade for Project 5.

The files you will need to run the tests are TestBinaryTree.java, ScoringRule.java, and WorthPoints.java. All of these files should be placed in the source code folder (the src folder in IntelliJ). Do not modify any of the provided files.

2 Getting Started with JUnit

Once all necessary files are added to the src folder, the JUnit testing libraries must be imported into the project. The simplest way to do this is to hover over the red import statements in the testing file then hit Alt+Shift+Enter as seen in the below photo.



An alternative way is to right click the import, click show context actions, then add 'JUnit4' to classpath.

3 Unit Test Overview

The only code you need to understand is in TestBinaryTree.java. Each test method in the files will be marked with an @Test flag. Tests that contribute to your score will have a @WorthPoints flag as well, indicating how many points that test is worth. Inside each method, there will be some statements calling methods of BinaryTree.java as well as some assertions. A test only passes if all of the assertions in the definition pass. For example, consider the following test:

```
public void ExampleTest() {
/*
Do some stuff here: initialize variables a, b, c, d, and e, call test
    methods,
```

```
etc.
*/
assertFalse(a);
assertTrue(b);
assertNull(c);
assertEquals(d, e);
}
```

This test will only pass if a is false, b is true, c is null, and d is equal to e. If any of those assertions are not met, the entire test fails.

4 Running the Tests

To run the tests in IntelliJ, click the green circle/play button to the left of the class declaration in TestBinaryTree.java. This will run all the tests in the class. To run a particular test only, click the green button next to that method. After running any tests, the failed assertions are underlined in red and a breakdown of what tests have failed is given. This should help give an idea of what's going wrong, but using a debugger is still recommended to fix difficult bugs.

Note: Although a test may fail, it does not definitively mean that particular method is incorrect. If there is a method that many other methods depend upon, it may result in more failed tests.