CSE331 HW6 Grading Key

Problem 1

Type params: type parameters not documented in Javadoc. (i.e. if your graph is defined as Graph<V, E>, then what is V and what is E?)
Wildcard: Misuse of wildcard generic type.
Extends/Super: Misuse of extends and/or super keywords in generic type definitions.
Generic type: Missing generic type where it is appropriate to have one.

Problem 2

Documentation for classes and methods (Javadoc for public classes and methods; private classes and methods can go either way)

> **No path:** Documentation for pathfinding method does not specify behavior if no path exists.
> **Doc:** Missing Javadoc comment for a class or method (or any comment above a private class or method).
> **Inner class doc:** Inner classes should be documented. Less documentation needed if declared as private and never exposed to the client.
> **Impl. details**: Javadoc comments should not contain implementation details.

All classes contain an AF and RI. Non-ADT classes should contain a comment stating as much in place of the AF/RI. (MarvelPaths2 isn't an ADT if contains only static methods. Comparator probably isn't an ADT, though this is more subtle.)

> **AF:** Missing abstraction function, or does not sufficiently map from concrete value (instance fields) to abstract value (all specfields). This is a difficult topic for most students, so please come talk to us at office hours if you're confused.
> **Path ordering:** Abstraction function of path class should specify that the nodes/edges are ordered.
> **RI:** Missing representation invariant.

Mostly for Pathfinding code

> **Comments:** Missing inline comments describing complicated code (e.g. build graph, pathfinding).
> **Var names:** Variable names need to be more descriptive.
> **Generic/Reusable Pathfinder:** Pathfinder should go in its own generic class for reusability. This will be an important code reuse issue for HW7.

Problem 3

Floats: assertEquals(double output, double expected, double delta) should be used to compare floating point numbers.

Test Coverage: Does not cover an adequate domain of inputs, including common cases for the path-finding algorithm as well as edge cases (nodes missing, no path, etc.).

Test Documentation: Non-descriptive test names or comments. The testing strategy or the goal of a particular test was not clear from reading the .test file.

Test Granularity: Tests do not represent coverage of a single domain. Each test (a method or a script file) should be short and test one specific case.

Impl vs Spec: Test cases in the implementation tests belonged in the specification tests. We did not mark off this time since we weren't clear about this in the past, but try to put your tests in the specification tests whenever possible.

Test Naming: Name of test does not indicate the condition being tested. e.g. "testStaffSuperhoes" or "testMarvel" - only describes the graph's abstract contents, not what is being tested.