ICS 52 – Introduction to Software Engineering
Final Exam – Winter, 2011

First Name: _____ Last Name: _____

1 - 2	3	4	5-6	Total

1. (12 points) In homework 4, you used (hopefully) a technique for selecting test cases called equivalence partitioning. This technique has several steps. List those steps, and briefly explain what is done in each one.

2. (10 points) The textbook distinguishes *dynamic* testing techniques (which execute code) and *static* techniques (which don't). Name and briefly describe one static technique described in lecture or the textbook. Describe a situation where this static technique is superior to a dynamic approach.

3. (20 points) You have been assigned to design test cases for black box testing of the two argument add method in java.util.Vector<E>. Here is part of its documentation:

```
public void add(int index, E element)
```

Inserts the specified element at the specified position in this Vector. Shifts the element currently at that position (if any) and any subsequent elements to the right (adds one to their indices).

(a) What is the input domain of the `add` method?

(b) What is a basis for dividing the input domain you described into subdomains?

(c) Using the basis defined in (b), specify 3 or 4 subdomains.

(d) For each subdomain from (c), give a test case input and the expected output.

4. (28 points, 4 points each) Define the following terms, as used in Software Engineering:

Test-driven development

Error-based testing

Control-flow coverage

Failure (in contrast to error, fault, mistake)

White-box testing

Oracle

Information Hiding

5. (15 points) Draw and label a diagram of the spiral software process model.

6. (15 points) The modularity of a design can be evaluated by the criteria of "cohesion" and "coupling."

Define cohesion, in the context of software modularity.

Define coupling, in the context of software modularity.

Is it best to have a lot of cohesion or little cohesion? A lot of coupling or little coupling? Explain your answers, referring to at least one software quality or software principle discussed in lecture or the textbook.