ICS 52 - Introduction to Software Engineering
Midterm Exam #2 – Winter, 2009

Last Name: _____ First Name: _____

1	2	3	4	5	6-11	Total

1. (15 points) The software engineering activity of testing is influenced by the general software engineering process. Select one software process model discussed in lecture or in the textbook, other than the waterfall model or the spiral model. What special issues in testing might arise when this model is being followed?

Many possible answers. Points off commonly for not making the connection between the process and testing clear, or for talking about testing issues (e.g. "it's time consuming") which hold true for all processes.

2. (20 points) Congratulations! You've been hired by SearchARoo, a company delevoping a new web based search engine that it hopes will be better than Google. Your title is Quality Assurance Engineer – Equivalence Partitioning Specialist. Your first task is to write a memo giving a high level overview of Equivalence Partitioning, the philosophy behind it, the steps involved, and at least one specific examples for each step of how it will work at SearchARoo. Restrict your memo to "black box" testing, and remember that you were hired because of your experience with homework 4 in ICS 52.

3. (15 points) We've discussed various desirable qualities in software. Name and define a desirable quality in a set of test cases. How could you ensure that a set of test cases actually had this quality – that is, how could you test the test cases?

   Some common errors:
   - **identifying a quality in software (not in test cases) and showing how it could be tested for – this was often indicated by our writing "Not T.C."**
   - **choosing a quality that is very difficult to achieve, such as "correctness" and then answering without acknowledging the difficulties**
   - **not being clear about how the ensuring works, e.g. suggesting we can ensure node coverage by using a control flow graph, but not saying how the process would work**

4. (12 points) The textbook says that an object (as in *object*-oriented programming) is a collection of three aspects:

   object = identity + state + behavior

   Explain what each of the three aspects are in Java terms. Can a Java `int` be considered an object, according to this point of view? Explain why or why not.

   (3 pts) identity: variable name or location in memory
   (3 pts) state: value of fields / member variables
   (3 pts) behavior: object's class's methods

   (3 pts) int? Can argue either way – it's not an object in Java, but it has state (a value), perhaps behavior (how it works with the +, - operators), and probably not identity (if 42 occurs twice in a program, they don't necessarily have separate identities)

5. (15 points) Select one of the principles of software engineering discussed in the lecture, and one of the architectural models presented in the textbook or in lecture. Briefly define each, and then explain how the principle you selected is exhibited in the model you selected.

   Principle and definition:

   5 pts. – 3 points off if something other than principle was selected

   Model and definition:

   5 pts. – 3 points off is something other than an arch. model was selected

   Explanation:

   5 pts. – full credit for making a clear connection

Answers appear in different orders on different copies

6. (3 points) The statements "We have to show something to our customer" and "We are judged by the amount of code written per person-month" and "We are pressed for time" are, according to the textbook, sometimes given as justifications for (choose the best answer)
   A. using the waterfall model.
   B. programming in Java.
   C. **not spending time on design.**
   D. writing elaborate class diagrams.
   E. eliminating white-box testing.

7. (3 points) Specialization in a UML class diagram usually corresponds to which keyword in Java?
   A. implements
   B. public
   C. private
   D. abstract
   E. extends

8. (3 points) A well-designed set of modules will usually have (choose one)
   A. loose coupling and high cohesion.
   B. loose coupling and low cohesion.
   C. tight coupling and high cohesion.
   D. tight coupling and low cohesion.

9. (3 points) In the Model-View-Controller design pattern, the job of the Controller is to (choose one)
   A. determine which model executes first.
   B. handle input actions.
   C. store the data.
   D. handle exceptions.
   E. control the flow of data between the various views.

10. (3 points) Which of the following activities would be associated with the upper left hand quadrant of the spiral model? (Choose one.)
    A. Prototyping and resolving risks.
    B. **Determining objectives.**
    C. Acceptance testing.
    D. Developing software requirements.
    E. Software product design.

11. (6 points) Define “testing oracle” as used in software engineering.
    A mechanism for determining the correct outcome of a test case.