University of Washington
CSE 403 Software Engineering
Spring 2011

Midterm exam

Friday, May 6, 2011

Name: _____

CSE Net ID (username): _____

UW Net ID (username): _____

This exam is closed book, closed notes. You have **50 minutes** to complete it. It contains 22 questions and 8 pages (including this one), totaling 100 points. Before you start, please check your copy to make sure it is complete. Turn in all pages, together, when you are finished. **Write your initials on the top of *ALL* pages.**

Please write neatly; we cannot give credit for what we cannot read.

Good luck!

Page	Max	Score
2	16	
3	9	
4	17	
5	19	
6	18	
7	10	
8	11	
Total	100	

1 True/False

(2 points each) Circle the correct answer. T is true, F is false.

1. **T / F** Showing your customer a mockup of the UI is one good way to get feedback while gathering requirements.

2. **T / F** For an announcement within your team, email is more appropriate than making the announcement during a meeting.

3. **T / F** When evaluating a UI via paper prototyping, the materials should have a crisp, professional appearance, to aid in obtaining good feedback.

4. **T / F** Cohesion refers to elements in the same module, whereas coupling refers to elements in different modules.

5. **T / F** Suppose you are able to find the true revealing sub-domains for a software system, and you design unit tests for it accordingly. Later, if you change your implementation, it is also necessary to update your tests so that they still cover the revealing sub-domains, which may have changed.

6. **T / F** When designing tests, if partitions are chosen perfectly, there is no point to testing boundary values near the edges of the partition.

7. **T / F** Glass-box tests designed for one implementation are valid to use when testing another implementation.

8. **T / F** In an implementation of the singleton pattern, there is no constructor.

2 Multiple choice

9. (3 points) Match each lifecycle model with its definition, by drawing a line connecting them.

(a) code-and-fix	(a) assess risks at each step; do most critical action first
(b) evolutionary prototyping	(b) build an initial small requirement spec, code it, then "evolve" the spec and code as needed
(c) spiral	(c) build initial requirement specs for several releases, then design-and-code each in sequence
(d) staged delivery	(d) standard phases (requirements, design, code, test) in order
(e) waterfall	(e) write some code, debug it, repeat (i.e. ad-hoc)

10. (3 points) A windowing system uses the notion of "damage" for what purpose?

    (a) to ensure correctness
    (b) to improve efficiency
    (c) all of the above
    (d) none of the above

11. (3 points) Which of the following are appropriate for a requirements document? Circle all that apply.

    (a) Multiple users will be able to log on without experiencing conflicts or slow load times.
    (b) The program will not have any bugs.
    (c) If the systems detects a major issue, it will save state and restart.
    (d) The user will be able to modify the location of the program's save location on their machine.
    (e) The program will disconnect the user if they enter an invalid value in any section.

3 Short answer

12. (4 points) The later a problem is found in the software development process, the harder it is to fix. Explain why, in one sentence.

13. (4 points) Give an example of a design pattern whose use is obvious from a class diagram but not from a sequence diagram. (Don't choose one that is built into (some) programming languages, such as inheritance.) Explain why, in 1 sentence.

14. (4 points) Give an example of a design pattern whose use is obvious from a sequence diagram but not from a class diagram. (Don't choose one that is built into (some) programming languages, such as iteration.) Explain why, in 1 sentence.

15. (5 points) Event-driven programming is appropriate for user interfaces and user interaction.

    - Give a *different* example domain.

    - In one phrase or sentence, state the characteristics of domains where event-driven programming is appropriate.

16. (5 points) Consider two components A and B. Two software engineers, Laurel and Hardy, measure the dependences between A and B. Laurel uses these dependences when computing cohesion, and Hardy uses these dependences when computing coupling. Is this possible, if both engineers are performing a sensible and useful computation? In 1–2 sentences, explain why or why not.

17. (6 points) In 1 sentence each, give two distinct reasons that you should not commit compiled code (such as `.o` or `.class` files) to a version control repository.

-
-

18. (8 points) It is cheaper and faster to fix known bugs before you write new code. Why? In one phrase or sentence each, give three reasons. Give reasons that are as different from one another as possible.

-
-
-

19. (8 points) After you find a bug but before fixing it, you should create a test case for it. In one sentence each, give three reasons that this is a good idea. Give reasons that are as distinct as possible.

    - _____
    - _____
    - _____

20. (10 points) Consider a wrapper whose implementation logs each call that occurs.

    In no more than 2 sentences each, explain when the wrapper should be considered a decorator (and why), and when that same wrapper should be considered a proxy (and why).

    - Decorator: _____
    - Proxy: _____

21. (10 points) Recall that the interning pattern guarantees that any two objects with the same abstract value are represented by just one concrete object. Answer each part in one sentence.

    (a) Give a usage pattern (or its characteristics) in which the interning pattern uses less memory, compared to not using it, and explain why.

    (b) Give a usage pattern (or its characteristics) in which the interning pattern uses more memory, compared to not using it, and explain why.

    (c) Give a usage pattern (or its characteristics) in which the interning pattern uses less time, compared to not using it, and explain why. Ignore effects that are really due to memory use, such as faster allocation.

    (d) Give a usage pattern (or its characteristics) in which the interning pattern uses more time, compared to not using it, and explain why. Ignore effects that are really due to memory use, such as thrashing.

22. (11 points) Explain the difference between a self-call, a synchronous callback, and an asynchronous callback. Give a concrete example of each. Draw a sequence diagram for each, and explicitly mark the self-call or callback in each (one marked call in each).

Self-call: Explanation: _____

_____

_____

Example: _____

Sequence diagram:

Synchronous callback: Explanation: _____

_____

_____

Example: _____

Sequence diagram:

Asynchronous callback: Explanation: _____

_____

_____

Example: _____

Sequence diagram: