Final Exam

CS 633 · League

Assigned 13th December; due 20th December 2004 at 5:00 PM

This is a take-home exam. You may use any books, notes, or web sites you wish, but you may not work with other students. If you use some text or ideas from a book or web site, please cite the source, and be sure to add some analysis in your own words. Please send your answers to `christopher.league@liu.edu` or bring a hard copy to the CS office. **The deadline is firm. I must submit your grades on Tuesday the 21st.**

1. In class, I showed how the "Chain of Responsibility" design pattern could be used to implement a very flexible web server with different kinds of content handlers (FileHandler, CodeHandler, NotFoundHandler). What are the advantages and disadvantages of this approach?

2. Below is a UML class diagram for a course registration system (figure 1). Modify and augment the diagram to account for the following facts:

   (a) Some majors are offered in conjunction with more than one department.

   (b) Students must have just one major, but they can *minor* in many different programs.

   (c) A major consists of some set of courses.

   (d) Students may be undergraduate or graduate, U.S. citizens or international.

   (e) Graduate students may tutor undergraduate students.

   (f) The University has multiple campuses, and each campus has several schools.

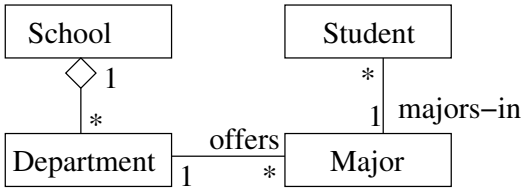

Figure 1: Class diagram for question 2.

3. Figure 2 on the next page contains a collaboration diagram for a word-processing application.

   (a) Create a sequence diagram that illustrates the same interaction.

   (b) Would any of these classes benefit from introducing a new *super* class? What would it be?

   (c) Draw a class diagram to model the classes involved in this interaction. Include the methods in each class. Include the super class, if any, from part (b).

4. What is the main difference between static and dynamic verification? Give one example of each kind.

5. What should be the primary goal of testing: (a) to demonstrate that the system works, or (b) to expose the faults in the system? How would each of these goals influence the test cases that you choose?

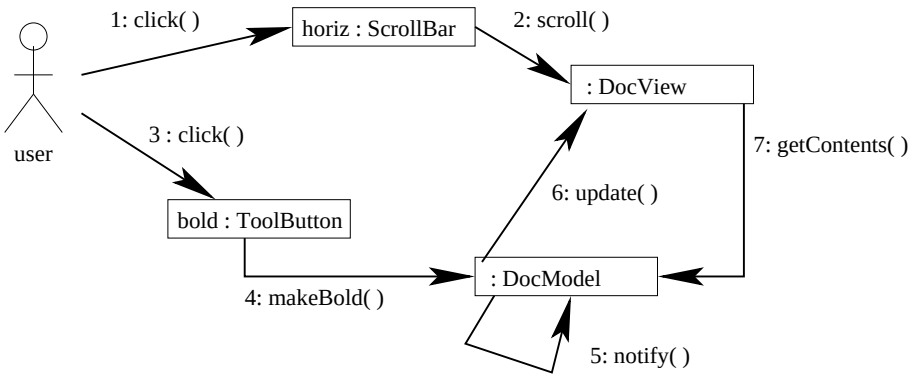

Figure 2: Collaboration diagram for question 3 on the preceding page.

6. As a test engineer for Boeing, you have been assigned to test the following pseudo-code. It has two parameters, `altitude` and `pitch`.[1]

```
FUNCTION autoPilot (altitude, pitch : integer)
BEGIN
    IF altitude > 10000 AND pitch < 0
    THEN RETURN pitch / 2
    ELSE IF altitude > 5000 AND pitch < 70
         THEN RETURN pitch * 0.3
         ELSE IF pitch < -25
              THEN RETURN (- pitch) / 2
              ELSE RETURN altitude + pitch
END
```

Your colleague suggests the following three test cases. Each case contains proposed inputs (values for `altitude` and `pitch`) and the expected output (return value).

altitude	pitch	return
10384	70	10454
986	–32	16
768	0	768

Do these three test cases produce good *coverage?* If not, suggest one or two additional cases that will make your testing more comprehensive. (Be specific: give proposed inputs and expected output.)

7. The first law of software evolution (page 470 in your book) states that "a program that is used undergoes continual change or becomes progressively less useful." Do you agree with this statement? Give a concrete examples of systems for which you think the law is and is not true.

8. While studying system *design,* we learned that reducing coupling between components can improve the overall quality of the system. Explain the impact of coupling on system *maintenance* in particular.

[1] It doesn't matter to this problem, but *altitude* is the distance of a plane off of the ground, and *pitch* is the angle of the nose with respect to the ground.