

Final Examination

Information and Computer Science 121

Professor Debra J. Richardson
Spring 1998

Name: _____

Signature : _____

Student ID: _____

Instructions

1. This exam is closed book, closed notes, closed neighbors.
2. The exam has 12 pages (don't panic ... there are lots of pictures).
3. Print your name and student ID number and sign the exam where indicated above. Print your name on each page of the exam.
4. For *true/false* questions, a statement is false unless it is entirely true; circle true or false in the left margin.
5. For the *problems*, draw your solution models neatly in the space provided.
6. For *very brief answer* questions, please respond in just a few words; write your answer neatly in the space provided.
7. For *short answer* questions, please be clear and concise in your answers; write your answer neatly in the space provided.
8. All answers are to be given on these pages. Use the back sides of the pages only if necessary, with a reference as to where any extra portion of the answer is located.
9. The maximum score for this exam is 100 points. Each true/false question is worth 1 point, problems have varying point values, each very brief answer is worth 2 points, and each short answer question is worth 4 points.
10. You have two hours to complete the exam – that is, until 12:30. Plan your time accordingly.

question type	true/false	problem	very short answer	short answer
# of questions	10	4	14	10
worth (%)	1/# = 10	22	2/# = 28	4/# = 40
grade (%)				

total grade	%
-------------	---

True/false questions (1 point each. Circle true or false in the left margin.)

- | | | | |
|------|-------|-----|--|
| true | false | 1. | The class model in OMT concerns itself with the events and actions in a software system. |
| true | false | 2. | White box testing does not consider the internal structure of a module when creating test cases. |
| true | false | 3. | An object can be composed of other objects. |
| true | false | 4. | Formal proof of correctness is a useful technique for validation. |
| true | false | 5. | Specification-based testing or implementation-based testing should be done, but there is no reason to do both. |
| true | false | 6. | Formal specification languages are not only useful in the requirements specification phase but also throughout the software lifecycle. |
| true | false | 7. | Testing activities can occur only after the code is implemented. |
| true | false | 8. | The purpose of a code walkthrough is to identify faults and correct them. |
| true | false | 9. | A makefile expresses the dependencies between files in a build. |
| true | false | 10. | Path coverage is a more thorough form of testing than edge coverage. |

Problems (varying points per question. Please provide diagrams in the space provided.)

11. Object-Oriented Analysis and Design (4 points)

You are tasked to design the software to support a computerized banking system with two ATMs. The banking system needs to maintain customer accounts and process transactions. Each customer may have up to four accounts. In conducting a transaction, a customer interacts with either a human cashier or an ATM to deposit, withdraw, or transfer funds between accounts. When using a human cashier, the customer must provide her name, account #, and mother's name. When using an ATM, the customer must provide her name, account #, and PIN.

Based on the system description above, identify the nouns and classify them into the following categories:

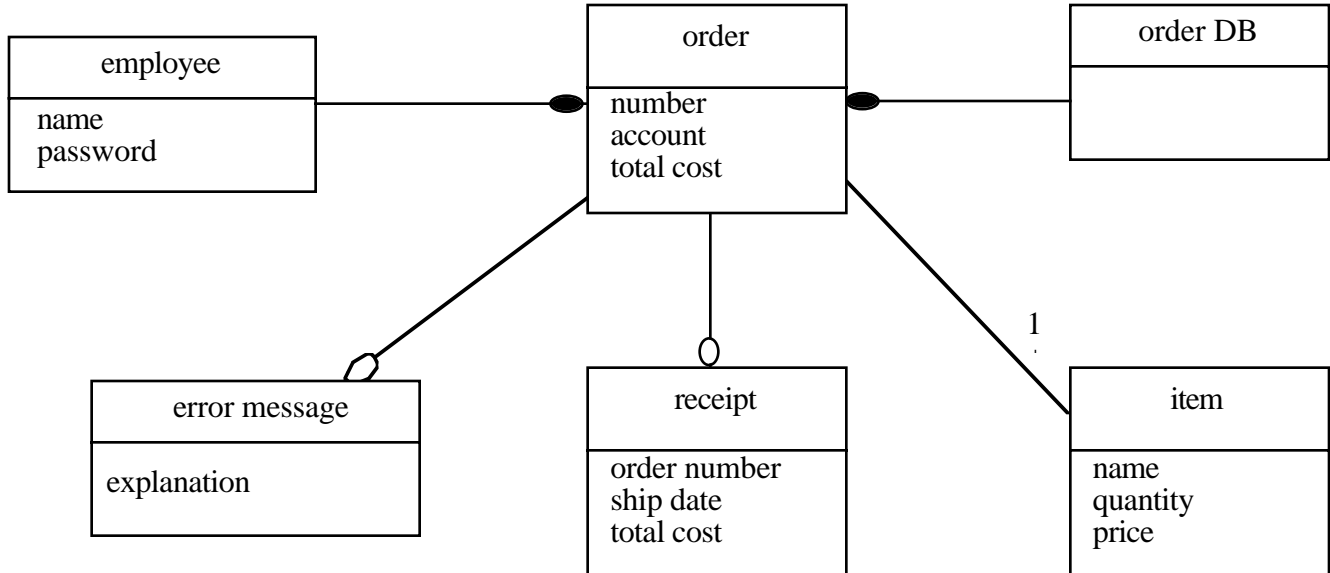
Good Candidates (possible Classes)

Abstract (possible Attributes)

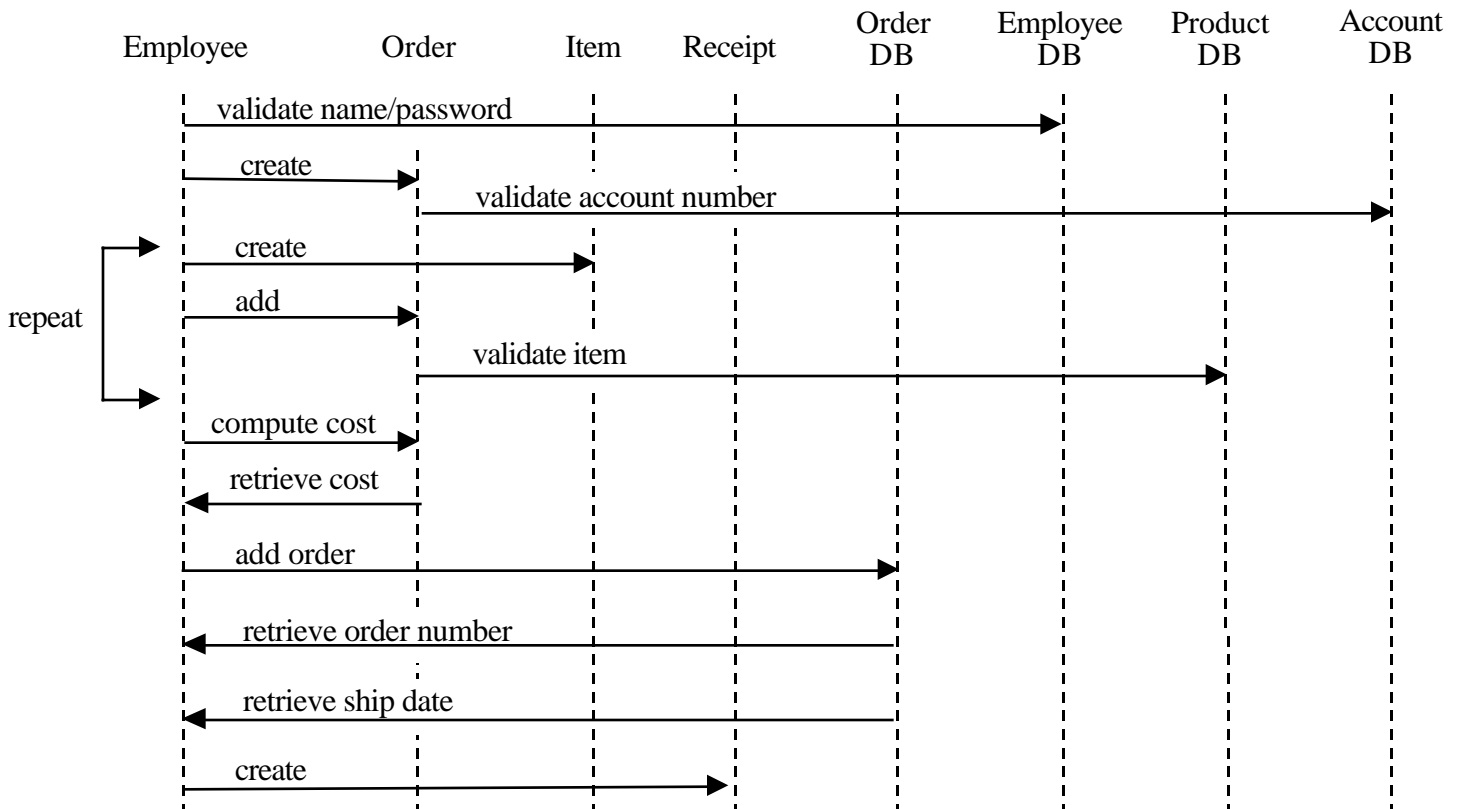
Out of Domain

12. Object-Oriented Analysis and Design (7 points)

Recall the following class model for the FastData example discussed in class:



Below is an event trace diagram for a successful order:



Object-Oriented Analysis and Design (continued)

In the space below, refine the class model above to be consistent with the dynamic model partially reflected above – i.e., introduce any additional classes (including any relevant aggregation and/or inheritance), and add operations and attributes to the classes as appropriate.

In your class model above,

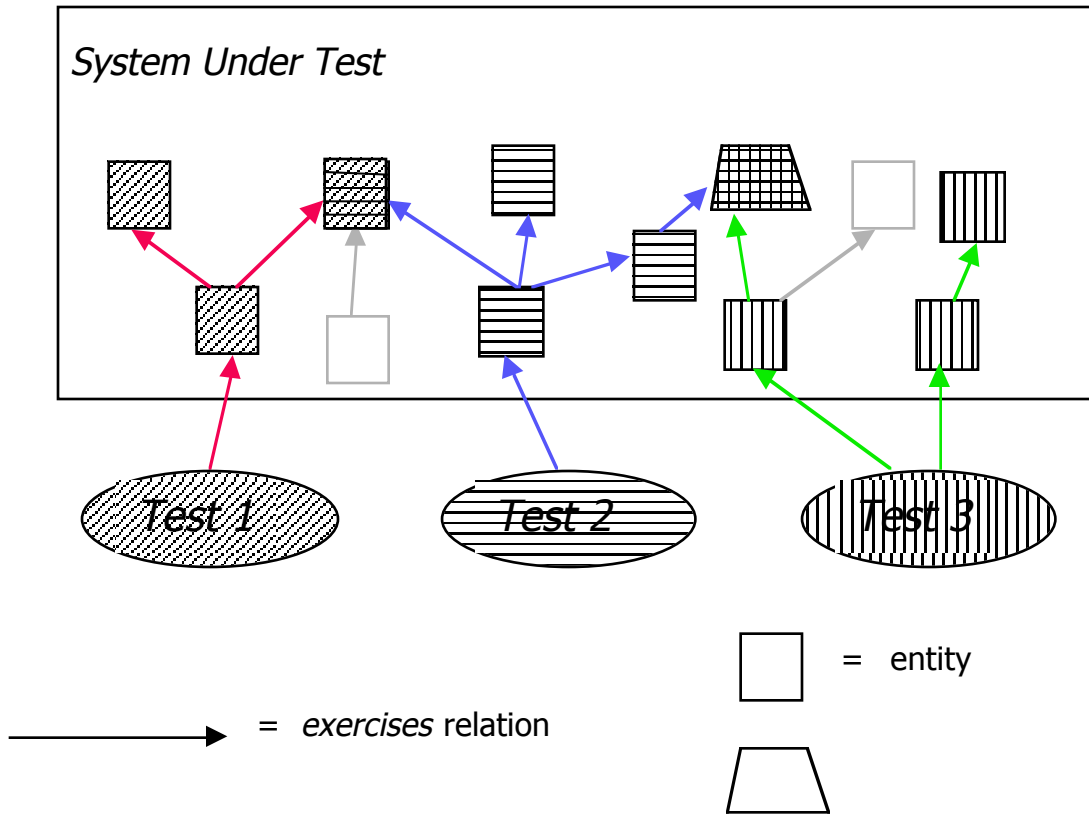
- a. Identify one occurrence of inheritance (e.g., class A inherits from class B).

- b. Identify one occurrence of aggregation.

- c. Identify one occurrence of polymorphism.

14. **Regression Testing (4 points)**

Given the dependence graph shown below, what tests should be re-executed if the odd-shaped module is modified?



Very brief answers (2 points each. Please answer each question in 1-6 words in the space provided. You will not be penalized for more words, as long as they are directly related to the question; but be aware that you should be able to answer the question in a few words.)

15. In *No Silver Bullet*, Brooks divides the difficulties of software engineering into *essence* (difficulties inherent in the nature of software) and *accidents* (difficulties that arise from current software production techniques). What are the names of the four inherent properties of the essential difficulties?

16. Why is it important to consider *selective* regression testing during maintenance?

17. What is the difference between a *fault* and a *failure*? Which one leads to the occurrence of the other? Does one always lead to the occurrence of the other?

18. _____ provides a good *test oracle* when doing *verification* via testing, while _____ provides a good *test oracle* when doing *validation* via testing

19. Acceptance and regression testing are two “phases” or “levels” of testing that are done after a system is delivered, what are the three levels of testing that take place before a system is delivered? List the three levels in the most appropriate order of their application.

20. One purpose of testing is to find faults in the software, name two others.

21. What aspects of a program or module are covered in the all-uses criterion that may not be covered in the all-edges criterion?

22. What is the primary purpose of conducting reviews and inspections throughout the software lifecycle and why should this be done?

23. One “phase” of a inspection is the actual conducting of the inspection. What two phases (activities) take place before actually conducting the inspection?

24. Why must you never delete configurations that represent released systems from a configuration management system?

25. How does the application of separation of concerns differ from the application of abstraction in Object-Oriented Analysis and Design?

26. If you have developed an object-oriented design, but are forced to implement in a programming language that is not object-oriented, what would you have to do?

27. Higher quality software is a major benefit of using formal specification languages. List two others benefits of their use.

28. Axiomatic specification is a basic type of behavioral specification, which is property-oriented. What two basic types of formal specification are model-oriented?

Short answers (4 points each. Please answer each question in two or three sentences in the space provided.)

29. Brooks wrote *No Silver Bullet* in 1986. What does Brooks' original *No Silver Bullet* argue?

Nine years later, with the publication of the 20th Anniversary edition of *The Mythical Man-Month*, Brooks revisits the *NSB* argument. In *No Silver Bullet Refired*, did he believe that his argument still held up or did he change his position? Explain Brooks' current (1995) position, what, if anything changed, and why he changed it or why he didn't.

30. Does developing more scenarios and adding event trace diagrams improve your capability to develop a good state transition diagram? If so, why? If not, why not?

31. Describe two advantages of doing Object-Oriented Analysis and Design over Structured Design. What is the difference in focus of the two design methods that leads to these advantages?

32. a. List the four phases of Object-Oriented Analysis and Design:

How do these phases relate to the following software development activities:

b. Requirements Specification:

c. System Modeling:

d. Architectural Design:

e. Structured Design:

f. Module Design:

33. What is the difference between functional and structural testing? Why are both needed?

34. It is generally accepted that no single technique is sufficient for adequate verification and validation of a complex system. Yet, a test manager must be careful to choose a cost-effective combination of testing techniques because of the limited resources available for testing. What guidelines could be used to decide what techniques to combine? Given these guidelines, what might you choose to combine?

35. What is the purpose of a regression testing process? How could *make* be used to automate this process?
36. If test criterion C2 subsumes criterion C1, and test set T2 satisfies C2.
- Will T2 satisfy C1? Why or why not?
 - Suppose that test set T1 satisfies C1. Must T1 be a subset of T2? Why or why not?
37. What activities are assisted or automated by a build management system and a configuration management system? In defining these activities, relate the terms build and configuration to each other and differentiate build management and configuration management relate from each other.
38. List four reasons why the Software Quality Assurance representative should be the chair of a code walkthrough. Three of the reasons explain why the author of the document, the author's manager, and the client should NOT be the chair of the code walkthrough.