Final Exam CISC 475/675 Fall 2004

True or False [2 pts each]:

1. (True/False) All software development processes contain at least separate planning, testing, and documentation phases.

2. (True/False) The main idea of the spiral life-cycle model was the introduction of risk management/mitigation as a primary organizing concept.

3. (True/False) In the Unified Process, each phase should have exactly 4 iterations.

4. (True/False) In the Unified Process, the four phases are Requirements, Analysis, Deisgn, and Implementation.

5. (True/False) Because the Chief Programmer Team approach is generally impractical, most organizational managers suggest replacing the chief programmer with both a team leader in charge of technical aspects and a nontechnical team manager.

6. (True/False) In Walkthroughs (as opposed to Inspections), the team is charged with documenting and correcting any faults found.

7. (True/False) A module has Procedural Cohesion if it performs a series of operations related by the sequence of steps to be followed by the product.

8. (True/False) The primary focus of the Requirements workflow/discipline is determining what the *client* wants.

9. (True/False) The primary output of the Requirements workflow/discipline are use case diagrams.

10. (True/False) The noun-extraction method is one technique used in the Analysis workflow/discipline.

11. (True/False) In general, Collaboration and Sequence diagrams can be used interchangeably.

12. (True/False) Responsibility-driven design is used to solve the method-assignment problem.

13. (True/False) Statement, Branch, and Path Coverage are common black-box testing techniques.

Short answer:

14. [2 pts] The main difference between a domain model and a design model is:

15. [2 pts] Name one thing that is better about top-down integration and one thing better about bottom-up integration.

16. [3 pts] Consider a software system to control an “intelligent” home. What are the most likely UML stereotypes for the following classes [chosen from the set *entity class*, *boundary class*, *control class*]?
    a. HotWaterHeater Class
    b. SecurityProcess Class
    c. FiveDayWeatherForcast Class

17. [3 pts] List three important components of a use case.

18. [2 pts] What is the difference between coupling and cohesion?

19. [2 pts] Give one argument FOR and one argument AGAINST formal correctness proofs for software.

20. [3 pts] Name three unique characteristics of XP (eXtreme Programming) teams.

21. [2 pts] Briefly describe the difference between CMM-2 and CMM-5.

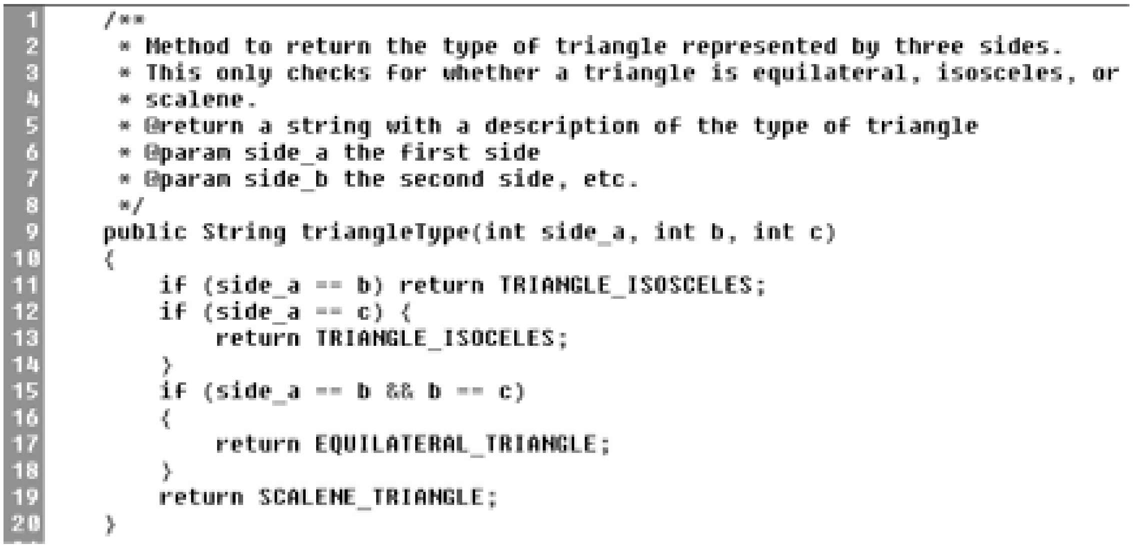
22. [2 pts] What is the main difference between waterfall lifecycle models and interarate-and-increment lifecycle models?

23. [2 pts] Briefly explain the cost of fixing a fault vs. the time at which it is detected.

24. [10 pts] Draw a simple domain model for the following use case involving a POS (Point Of Sale) terminal:

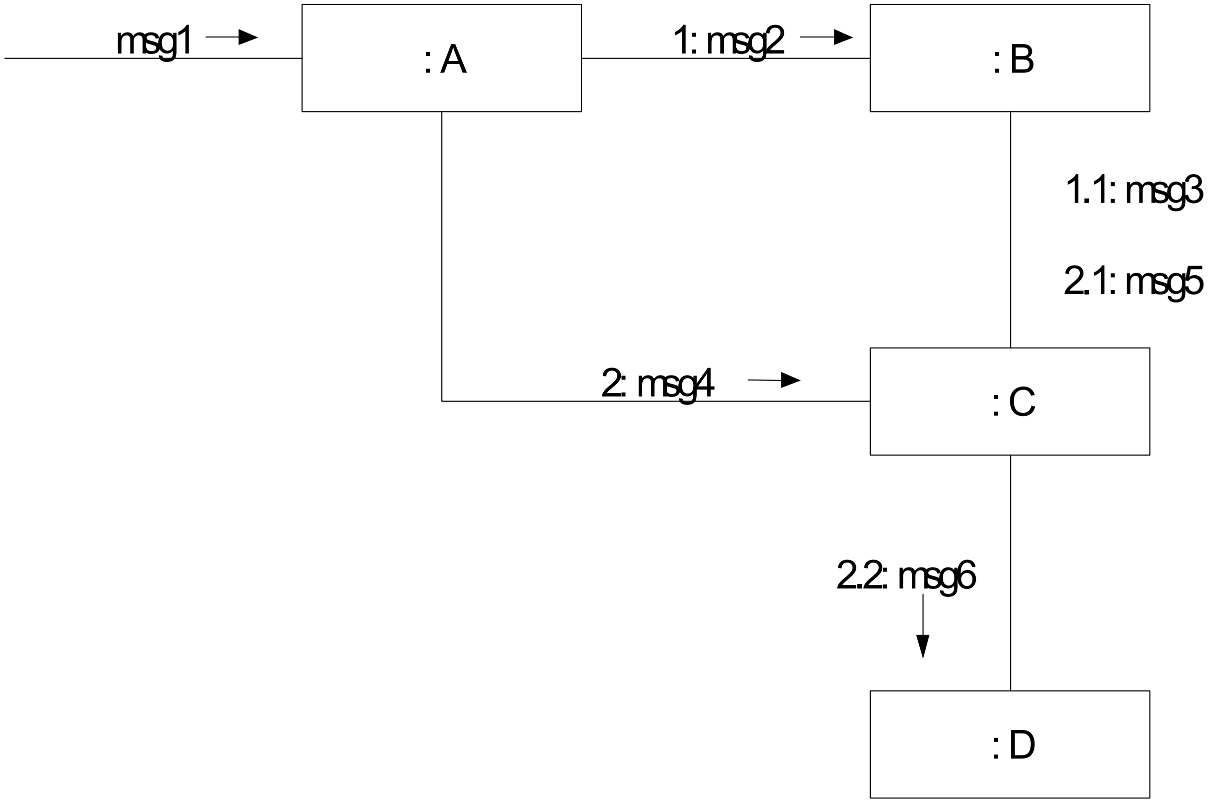
Main Success Scenario (or Basic Flow): 1. Customer arrives at a POS checkout with goods and/or services to purchase. 2. Cashier starts a new sale. 3. Cashier enters item identifier. 4. System records sale line item and presents item description, price, and running total. Price calculated from a set of price rules. Cashier repeats steps 2-3 until indicates done. 5. System presents total with taxes calculated. 6. Cashier tells Customer the total, and asks for payment. 7. Customer pays and System handles payment. 8. System logs the completed sale and sends sale and payment information to the external Accounting (for accounting and commissions) and Inventory systems (to update inventory). 9. System presents receipt. 10.Customer leaves with receipt and goods (if any).

25. [10 pts] Assume you were going to review the following code. You are to inspect it for consistency in naming, formatting, and understandability of style. You also are going to review it for correctness. Identify all of the things you find that you would raise during a code review. Identify each item by specifying the line number (if appropriate) and the problem.



```
 1    /**
 2     * Method to return the type of triangle represented by three sides.
 3     * This only checks for whether a triangle is equilateral, isosceles, or
 4     * scalene.
 5     * @return a string with a description of the type of triangle
 6     * @param side_a the first side
 7     * @param side_b the second side, etc.
 8     */
 9    public String triangleType(int side_a, int b, int c)
10    {
11        if (side_a == b) return TRIANGLE_ISOSCELES;
12        if (side_a == c) {
13            return TRIANGLE_ISOCELES;
14        }
15        if (side_a == b && b == c)
16        {
17            return EQUILATERAL_TRIANGLE;
18        }
19        return SCALENE_TRIANGLE;
20    }
```

26. [10 pts] Consider the following diagram. What kind of diagram is it? Explain what it shows.



27. [10 pts] Consider a system with the following 4 domain classes: an Order, representing a full customer order, an OrderLineItem representing one line of the full order, a Product, representing something that a customer might purchase as a part of an order, and a class representing the customer themselves.

    Draw a UML sequence diagram representing the following scenario: We have an order and are going to invoke a command on it to calculate its price. To do that, the order needs to look at all the line items on the order and determine their prices, which are based on the pricing rules of the order line's products. Having done that for all the line items, the order then needs to compute an overall discount, which is based on rules tied to the customer. Full credit for an object-oriented, distributed solution (where the Order passes off the price-calculation to the line, and the line to the Product...)

Two Points each:

28. How are a *workflow*, an *artifact*, and a *baseline* related?

29. What are the differences between *walkthroughs* and *inspections*?

30. Give an example of a nonfunctional requirement that can be handled without having detailed knowledge about the target software product.

31. Why are the attributes of classes but not the methods determined during object-oriented analysis?

32. Define any two of the following design patterns: [*Information Expert, Creator, Controller, Low Coupling, High Cohesion, Polymorphism, Pure Fabrication, Indirection, Don't Talk to Strangers*]