

ICS52 - Introduction to Software Engineering  
Second Midterm Exam – Winter, 2006

Last Name: \_\_\_\_\_ First Name: \_\_\_\_\_

Row: \_\_\_\_\_ Seat Number: \_\_\_\_\_

1. (15 points, 5 points each) Define the following terms, as used in software engineering:

A. Software Architecture.

**Top level decomposition of a system into major components together with a characterization of how these components interact. (p. 254)  
Also see p. 255.**

B. Information Hiding.

**Each module has a secret; or, Implementation is hidden.  
1 point off for "hidden from human user." See page 303.**

C. Functional Decomposition.

**A design approach which applies divide and conquer to break a large system into subfunctions.  
See pages 317 – 320.**

2. (12 points) Joel Spolsky writes, "Design, for my purposes, is about making tradeoffs." Select two software qualities discussed in lecture or in the textbook (but not "high-quality", "on-time", or "inexpensive"), and describe how pursuing one quality might be detrimental to achieving the other. Discuss this tradeoff in the context of a specific software system, ideally one that has been mentioned in the textbook or lecture.

**2 points for naming two qualities**

**5 points for discussing the trade-offs**

**5 points for putting this in the context of a specific system**

3. (12 points) Briefly describe two of the three main purposes of software architecture.

#1

**See page 254.**

**It is a vehicle for communication among stakeholders**

**It captures early design decisions.**

**It is a transferable abstraction of a system**

#2

4. (15 points) In Homework 2, you wrote Class Interface Specifications. Describe how writing these exemplifies each of the following design principles.

abstraction

**Concentrates only on functionality and usage of methods, and not on their implementations.**

modularity

**Reduces coupling (modules don't rely on each others internals).  
Since modules don't rely on the implementation of others, it is less likely that change in one will require change in another.**

information hiding

**Encourages modules to hide design decisions.**

5. (24 points) Consider an exciting new system called Multi-Language Instant Messaging (MLIM). MLIM works like conventional Instant Messaging, except that each user selects his/her language from a drop-down box. Outgoing and incoming messages are automatically translated to or from the language of the receiver or sender, if necessary. Select two architectural styles described in the book or in lecture, write down their names, and for each style draw a diagram (*not* a UML class diagram) showing the MLIM architecture following that style.

Style #1: \_\_\_\_\_

**General rubric, 12 points each; 3 for naming the style, 9 for the diagram.**

---

Style #2: \_\_\_\_\_

6. (10 points) According to the Scientific American article "Command and Control," ATAMS was successful because it "combined several techniques that were shown years ago to produce better software faster." Name and briefly describe two techniques mentioned in the article, and explain which software engineering qualities, principles, or process models each technique illustrates.

1. First technique:

2. Second technique:

**[You had to give techniques alluded to in the article.]**

**Displays and user interfaces were built first, rather than last. -- separation of concerns, user-friendliness**

**The system was split into small segments and the riskiest ones were done first. -- anticipation of change, modularity**

**Off-the-shelf software and large sections from other systems were used. -- reusability, modularity**

**Programmers peer-reviewed one another's designs and code, catching more than 200 major design errors while they were still easy to fix. -- understandability, incrementality**

**The engineers had to perfect each segment before moving on. -- correctness, verifiability, incrementality**

**Frequent contact with users. -- user-friendliness, validity of requirements, possible visibility**

**Will be updated once a year, rather than replaced once a decade. -- incrementality, maintainability, anticipation of change**

**First in a product line of related systems. -- reusability, anticipation of change, planning for a software family**

**5 points each, 3 for describing the technique, 2 points for the qualities/principles**

7. (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. **not spending time on design.**
  - B. using the waterfall model.
  - C. programming in Java.
  - D. writing elaborate class diagrams.
  - E. using information hiding.
8. (3 points) Specialization in a class diagram usually corresponds to which keyword in Java?
- A. **extends**
  - B. implements
  - C. public
  - D. private
  - E. abstract
9. (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.
10. (3 points) In the Model-View-Controller design pattern, the job of the Controller is to (choose one)
- A. **handle input actions.**
  - B. determine which model executes first.
  - C. handle exceptions.
  - D. control the flow of data between the various views.