

ICS 52 - Introduction to Software Engineering
Midterm Exam #2 – Fall, 2007

Last Name: ___K E Y ___ First Name: _____

1. (10 points) Why is the design stage of software development particularly difficult?

For full credit, the response had to mention something particularly difficult about the design stage, e.g. it's difficult to test a design; design diagrams are not compilable; design requires thinking at an abstract level; a design language has to be learned. About half credit for just describing the design phase, or identifying difficulties that apply to all aspects of software engineering.

2. (20 points) According to the Scientific American article "Command and Control," system development at the NORAD Command Center had many years of failures, followed by significant successes.

Select one software principle or quality described in lecture or in the textbook, and explain how ignoring that principle or quality contributed to the failures. Be specific about the problems.

10 points for each part; full credit only for responses that discussed specific features of the systems described in the article.

Select one software principle or quality described in lecture or in the textbook, and explain how following that principle or achieving that quality contributed to the successes. Be specific about what went right.

3. (20 points) Define “information hiding” in the context of software engineering. Give a specific example of information hiding in a design for the Schocr system. (The design can be similar to what you did for homework 2, or can be different.)

10 pts. for definition: “one module encapsulates information or design decisions” or something along those lines.

10 pts. for the example.

4. (20 points) Dijkstra says, “Testing can show the presence of bugs, but not their absence.”

- Why is this true about software testing?

10 pts. for an answer like “there are too many input cases to test all of them, and any one could reveal a bug in the software.”

- If we replace the word “bugs” with “defects” or “flaws,” is this claim just as true about testing the strength of fishing line, the color consistency of a series of art books, or the purity of a flute’s sound as it is about testing software correctness? Explain your answer.

10 pts. Full credit for answers that addressed what is different between software and these (analog, continuous) examples.

5. (30 points) Congratulations! You have just been hired as Software Architect for a new system called Geographical Information Assistant (GIA). GIA allows its users to type in a question such as “What is the capital of France?” or “What are the three biggest rivers in California?” or “Which countries border Paraguay but not Chile?” The questions can be asked in English, French, or Spanish. GIA can be accessed through a web page or via cell phone text messages. GIA then queries several databases, both public (on the WWW) and private (subscription only) to retrieve the information. The response is presented both as text in the language of the query, and as a map. Two programmers at your new company are arguing over what is the best software architecture for GIA. One says the Repository style is best, and the other says a Layered style is best. Discuss the pros and cons of these two approaches for this specific application, and for each approach draw a diagram (*not* a UML class diagram) showing the GIA architecture following that style. Make sure your diagrams are clearly labeled and clearly show a high-level design for GIA using each architectural style.

Repository style:

15 pts. for each style, about half for the diagram and half for the pros and cons.

Layered style: