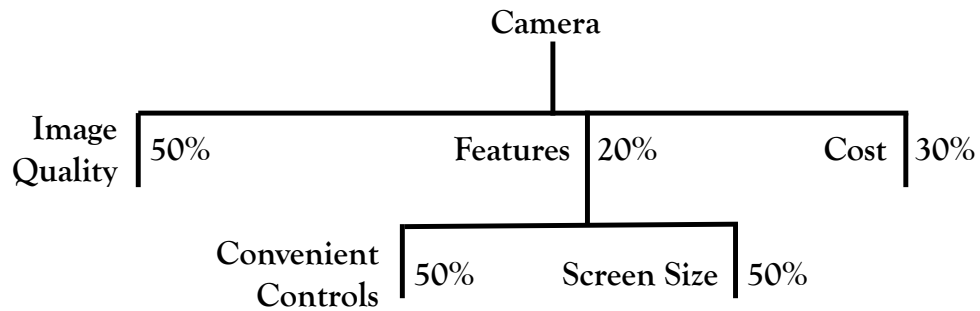


SIXTH QUIZ

You have 15 minutes from the start of class to complete this quiz. Give partial answers if you can't give complete ones. Read the questions with care; work with deliberate speed. Don't give us more than we ask for. The usual instructions apply. Good luck!

Problem 1 (11 points)

Below is a relevance tree for cameras. Fill in the six blanks to calculate the ratings of each alternative. Showing your work might increase your chances of getting partial credit if your arithmetic is wrong.



Overall relevance of each node: _____ _____ _____ _____

Photon D1350X	8	8	10	6	_____
Thumtax Q137	10	4	6	8	_____

Ratings of alternatives

(a) (7 points) Fill in the six blanks to calculate the ratings of each alternative. Showing your work might increase your chances of getting partial credit if your arithmetic is wrong. Finally, according to this relevance tree, which camera should you choose?

(b) (1 point) Which of the two cameras has the larger screen? (In digital cameras, larger screens are better.)

(c) (2 points) One of the two cameras costs \$750; the other costs \$900. According to the tree above, which camera has which price?

Problem 2 (4 points) Your web site gets 1,000 “hits” (visitors) each hour from midnight to 6:00 a.m., 2,000 hits each hour from 6:00 a.m. to noon, and 4,000 hits each hour from noon to midnight. What’s the expected number of hits in an hour (picked randomly out of the 24-hour day)?

Problem 3 (10 points)

In class we discussed the “software crisis”—the observation that software projects often take longer than expected and cost more, often fail entirely, and almost invariably produce results that still have bugs—and three different approaches to “solving” the crisis: Structured (“gotoless”) programming, N-version programming (“design diversity”), and formal verification (program proofs). Fill in five cells in the table below; you may leave any four cells empty (and one cell doesn’t have a good answer). Think carefully before you write; try to make your answer complete but small enough to fit in the space provided.

	Brief definition or description	One reason why it didn’t “solve” the software crisis (other than “It’s too expensive and time-consuming”).	One situation where this approach is appropriate or useful today, or one useful concept or technique that this approach produced.
Structured (“gotoless”) programming			
N-version programming (“design diversity”)			
Formal verification (program proofs)			

