

ANSWER ANY FIVE QUESTIONS, ALL QUESTIONS HAVE EQUAL WEIGHT

Questions attempted in excess of 5 will not be marked.

CLOSED BOOK and NOTES, ONE 8 ½ X 11 AID SHEET ALLOWED.

Write legibly; unreadable answers are not answers.

1. Define *Cyclomatic Complexity*. What is the significance of Cyclomatic Complexity for software testing?
2. As manager for software testing at a large software house you have to make a decision on how the fixed budget for testing should be split between performing *Code Inspections* and *Unit Testing*. What are the important factors that you will consider in making this decision? What do you think is a reasonable allocation of effort between these two activities?
3. Assume that you are managing a large software project. You thought the project was proceeding well until a simulation of the proposed design appeared to show that the system would not meet its performance objectives. Your programming team proposes to improve system performance by *increasing* the coupling between modules in the system through the use of shared global data. How would you respond to this situation and to your team's proposal?
4. You are doing Risk Analysis for a proposed software project. One risk that you have identified is that over half of the programmers assigned to the project are *part timers*. i.e. over 50% of each individual's time is committed to some other project. What steps would you recommend to compensate for this risk before and during the project?
5. Describe the software engineering problems that *Software Configuration Management Tools* are designed to solve.
6. You and your team members were so successful in the course project that you decided to form a small software house specializing in the design and implementation of high quality custom software. A company this small can't afford a separate Software Quality Assurance Department. Describe some steps that you could take in the context of a very small software company to guarantee that the software you deliver to customers is of very high quality.
7. A software designer once said:

"Defining interfaces is the most important part of system design"

Do you agree or disagree with this statement? Justify your answer.