**CSC 179**

**- Software Testing and Quality Assurance -**

**Summer 2024**

**Final Exam - Study Guide -**

1. List and briefly explain types of systems
2. Explain the meaning of a “System”, System Engineering and System Modeling
3. What are the differences between the following models: - SCD, DFD, UML Use Cases
4. How System Thinking can be helpful in software V & V
5. What is the essence of the Iceberg model?
6. List some of the process principles
7. Describe Software Quality in the context of V & V
8. What does V & V encompasses?
9. What does this mean “non-conformance is a defect “
10. What is the difference between Software Verification and Software Validation?
11. How test engineers decide when to use verification vs validation
12. Is this a true statement – “Testing is a process of executing a program with the intent of finding defects?”
13. What are the 2 main purposes of testing and briefly explain the challenge in each one
14. what are some of the major Software Testing challenges
15. How does a test engineer know “When to Stop Testing?”
16. Briefly compare some of the “Testing Metrics” that we discussed in class
17. What is the Software Testing Process and its four phases-?
18. What are some of the models that can be used to model the software under test?
19. What are the 2 criteria for “Test Case Design?”
20. List the major differences between Black Box Testing & White Box Testing
21. Explain Equivalence Partitioning/Classes with one example
22. What are some of the Equivalence Partitioning/Classes challenges?
23. Is Boundary Value Analysis the same as Equivalence Partitioning/Classes?
24. Is Error Guessing an effective V & V technique?
25. What are the 4 areas that White Box Encompasses?
26. Give an example of for the following: Statement Coverage -Decision Coverage - Condition Coverage
27. What is the difference between Decision Tables and Cause Effect Graphing
28. List some examples of Static Testing
29. What is a Flow Graph and how it can be used
30. What is Cyclomatic Complexity?
31. What is an infeasible path and can it be resolved
32. Is Unit testing effective – why and why not?
33. Briefly compare Integration testing to Unit testing
34. How do test engineers decide between using Bottom-up or Top-down integration?
35. Why adopt a Shift Left strategy
36. What is defect prevention?
37. What would be the key success factor for requirement-based testing?
38. Briefly explain the key challenges in model based testing
39. What is the effect of coupling and cohesion on design and testing
40. How sequence diagram can be sued for design and testing
41. What is the key difference between system testing and acceptance testing
42. In code reviews – what is meant by “what is wrong vs what is missing”
43. Why testing built on the concept of “risk based” approach
44. Why testing engineers should consider using test maturity models
45. What is the key focus of Design Thinking
46. Test to pass vs test to fail – what is the difference
47. Why test engineering use metrics to asses testing progress? Explain with an example
48. What is meant by reliability estimation and why it is important
49. What is the concern with “Equivalence Classes” as black box testing method
50. Testing NFRS – explain why it is difficult.