CS 422 Software Engineering Principles
Study Questions (Ch1-10 Sommerville)
Including some miscelaneous (Miscel) materials covered in lecture or homework (HW)

1. (Miscel) What is the difference between fault avoidance and fault tolerance?

2. (HW Article) What is the difference between black box testing and white box testing?

3. (HW Article) Explain the terms structural, random testing and equivalence partitioning.

4. (Miscel but see ¶24.4 in Sommerville) What is *Cleanroom* software development?

5. (Miscel) What are the major components of a context diagram (give an example if you need to)?

6. (Miscel but see Chap 22) Define the terms (a) Verification and (b) Validation.

7. (Miscel Video HW) Define what is a process (name three things and what are they).

8. (Miscel) What are standards and why are they important in software engineering?

9. (Miscel) What are the three sections of a progress report. Why are progress reports important in software engineering?

10. (Chap 1) Fill in the blanks of the following sentences:
    Software Engineering is concerned with theories, _____ and tools which are needed to develop the _____ for computer systems (e.g., aerospace, avionics, telecommunications, government, health case, etc.). Different from other engineering disciplines because it is not constrained by materials governed by _____ laws or by manufacturing processes. Software Engineers model parts of the _____ world in software. The goal is to produce practical software solutions in a _____ effective way. Products that are reliable, robust, _____, flexible and maintainable.

11. (Chap 1) What are the four major partitions in the Spiral model of the Software Life Cycle?

12. (Chap 1) Name three strategies that are used for risk resolution?

13. (Chap 1) Describe the five Process models reviewed in Sommerville and their relation to process visibility?

14. (Chap 2 + Miscel) What are the three major facets (influences) in system reliability engineering?

15. (Chap 2) Name the seven steps in System Engineering, and identify the step that uses the 'Big Bang' approach.

16. (Chap 3) In project management, what are five examples of outputs from project planning?

17. (Chap 3) What is an important output from developing a activity network?

18. (Chap 3) Define what is a project milestones.

19. (Chap 3) What is the critical distinction between a project milestone and a deliverable?

20. (Miscel) Name two subsections of the paragraph named "Specific Requirements" in an SRS based on the IEEE Standard.

21. (Chap 3) Well-managed projects sometimes fail. Badly managed projects _____fail.

22. (Chap 3) Project management is a _____ _____ from initial concept through to system delivery.

23. (Chap 3) The project plan contains the following sections (i.e., is structured accordingly) *Introduction, Project organization,* _____ _____*, Hardware and software resource requirements, Work breakdown, Project schedule,* Monitoring and reporting mechanisms.

24. (Chap 3) The scheduling problem is frought with pit falls. Here are some basic rules of guidance: Productivity is <u>not</u> _____ to the number of people working on a task; Adding people to a late project makes it _____ because of communication overheads; The unexpected _____ happens.

25. (Chap 3) In your own words, describe the term critical path based on the activity network described in Chapter 3 of Sommerville.

26. (Chap 4/5) Name the two main characteristics and/or purposes for the requirements traceability matrix.

27. (Chap 4) Briefly describe what is meant by requirements validation.

28. (Chap 4) Briefly describe what is meant by requirements evolution.

29. (Chap 5) What is the essential difference between viewpoint-oriented analysis and method-based analysis.

30. (Chap 6) A model is an abstracct view of a system which _____. Complementary system models can be developed that present _____ information about the system.

31. (Chap 6) Give a brief definition of the following three types of abstract system models:
    a. Data-flow models

    b. Semantic models

    c. Object models

32. (Chap 7) Explain the differences (at least three) between Requirements *Definition* and Requirements *Specification*.

33. (Chap 7) Explain what is meant by non-functional requirements (give at least three examples).

34. (Chap 8) Name and define all the different kinds of SW Prototyping?

35. What is Transformational development (Chap9)?

36. What is a functional abstraction (Chap9)?

37. Formal specifications are expressed in a mathematical notation with precisely defined (three terms) (Chap9)?

38. Name the phases and ordering of specification and design (Chap9).

39. What is system modeling versus architectural design (Chap9)?

40. What classes of software are difficult to specify using current techniques (Chap9)?

41. What are the advantages of formal specification (name four) (Chap9)?

42. Give the seven myths of formal methods (Chap9).

43. According to the author Sommerville, how should the development costs profile change going from a development process that does not use formal specification to one that does (Chap9)?

44. Describe the idea of pre- and post-condition when used in a software specification (Chap9).

45. What is a predicate and how is it used in the context of software specifications (Chap9)?

46. What is the basic difference between algebraic and model based approaches to formal specification to software systems? Give an example of a language from both domains which operates in the sequential and the parallel worlds of software systems (Chap9).

47. Formal specification forces an analysis of the system _____ at an early stage (Chap9).

48. Formal specifications are precise and _____ (Chap9).

49. Formal specifications are most applicable in the development of _____ - _____ systems because of there inherent high cost (Chap9).

50. Functions can specified by setting out pre and post conditions for the function. However, this approach does not scale up to _____ or medium sized systems (Chap9).

51. (Chap10) Algebraic specifications of a system may be developed in a systematic way as follows: (1) specification structuring, (2) specification naming, (3) operation selection, (4) informal operation specification, (5) _____ definition, and (6) axiom definition.

52. In a structured specification the following rules apply (mark true / false for each one):

    Specifications should be constructed in a structured way. Other specifications should be reused whenever possible (T___/F___)

    Specification instantiation. A generic specification is instantiated with a given sort (T___/F___)

    Incremental specification. Use simple specifications in more complex specifications (T___/F___)

    Specification enrichment. A specification is constructed by inheritance from other specifications (T___/F___)

53. (Chap10) Algebraic specification is particularly appropriate for _____ interface specification.

54. (Chap10) Algebraic specification involves specifying _____ on an abstract data types or object in terms of their inter-relationships.

55. (Chap10) An algebraic specification has a signature part defining _____ and an axioms part defining _____ .

56. (Chap10) Formal specifications should have an associated _____ description to make them more readable.

57. (Chap10) Algebraic specifications may be defined by defining the semantics of each inspection operation for each _____ operation.

58. (Chap10) Specification should be developed incrementally from _____ specification building blocks.

59. (Chap10) Errors can be specified either by defining _____ error values, by defining a _____ where one part indicates success or failure or by including an error section in a specification.

60. (Chap10) Given the 4 parts of a algebraic spec. structure, what does each part contribute?

    (a) Introduction

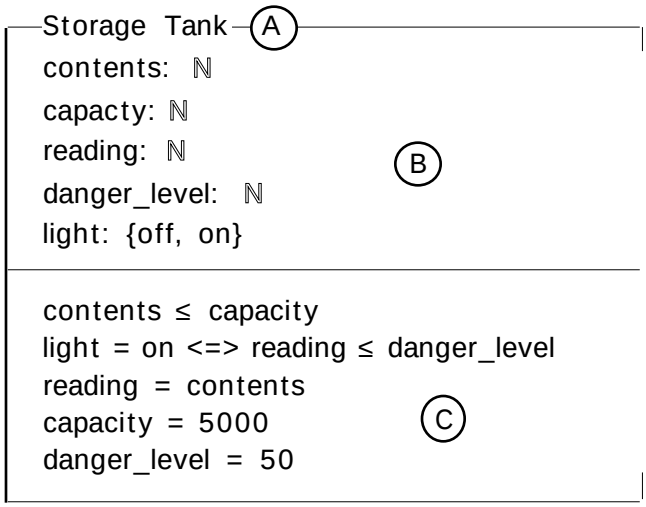
    (b) Informal description

    (c) Signature

    (d) Axioms

61. (Chap 10) An algebraic specification has a _____ part defining syntax and an _____ part defining semantics.

62. (Chapt 10) Given the two different specifications shown below answer the following questions: (a) name the two types (left one) _____ (right one)_____. Name the three parts of the specification on the left and the four parts of the specification on the right.


Storage Tank
A
contents: ℕ
capacty: ℕ
reading: ℕ
danger_level: ℕ
light: {off, on}
B
contents ≤ capacity
light = on <=> reading ≤ danger_level
reading = contents
capacity = 5000
danger_level = 50
C

A._____

B._____

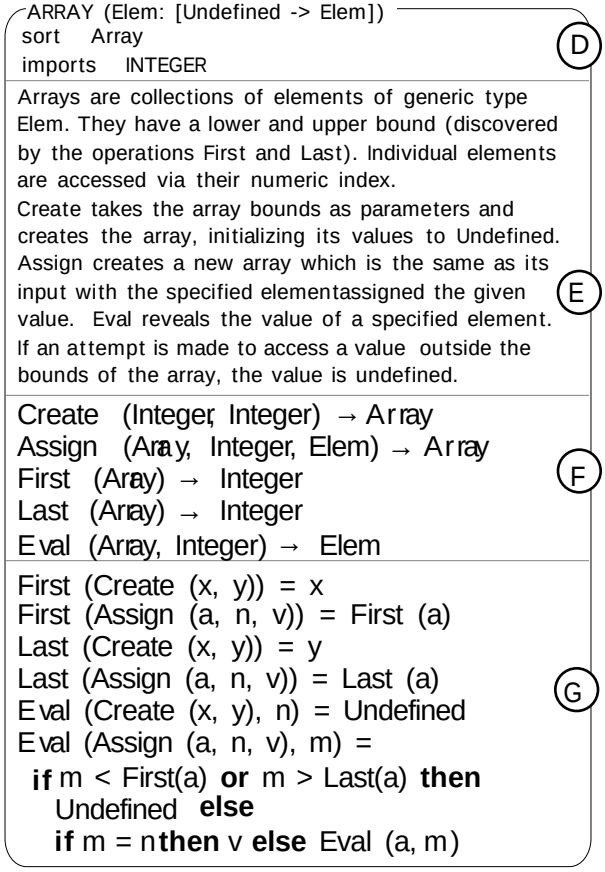
C._____

D._____

E._____

F._____

G._____.


ARRAY (Elem: [Undefined -> Elem])
sort Array
imports INTEGER
D
Arrays are collections of elements of generic type Elem. They have a lower and upper bound (discovered by the operations First and Last). Individual elements are accessed via their numeric index.
Create takes the array bounds as parameters and creates the array, initializing its values to Undefined. Assign creates a new array which is the same as its input with the specified elementassigned the given value. Eval reveals the value of a specified element. If an attempt is made to access a value outside the bounds of the array, the value is undefined.
E
Create (Integer, Integer) → Array
Assign (Array, Integer, Elem) → Array
First (Array) → Integer
Last (Array) → Integer
Eval (Array, Integer) → Elem
F
First (Create (x, y)) = x
First (Assign (a, n, v)) = First (a)
Last (Create (x, y)) = y
Last (Assign (a, n, v)) = Last (a)
Eval (Create (x, y), n) = Undefined
Eval (Assign (a, n, v), m) =
if m < First(a) or m > Last(a) then
Undefined else
if m = n then v else Eval (a, m)
G