

1. What is Transformational development (Chap9)?
2. What is a functional abstraction (Chap9)?
3. Formal specifications are expressed in a mathematical notation with precisely defined (three terms) (Chap9)?
4. Name the phases and ordering of specification and design (Chap9).
5. What is system modeling versus architectural design (Chap9)?
6. What classes of software are difficult to specify using current techniques (Chap9)?



13. Formal specification forces an analysis of the system \_\_\_\_\_ at an early stage (Chap9).
14. Formal specifications are precise and \_\_\_\_\_ (Chap9).
15. Formal specifications are most applicable in the development of \_\_\_\_\_ - \_\_\_\_\_ systems because of their inherent high cost (Chap9).
16. Functions can be specified by setting out \_\_\_ and \_\_\_ conditions for the function. However, this approach does not scale up to \_\_\_\_\_ or medium sized systems (Chap9).
17. Name the four major pieces of a (algebraic) specification structure (Chap10).
18. 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 (Chap10).
19. In a structured specification the following rules apply (mark true / false for each one) (Chap10):
- \* 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\_\_\_)
20. Algebraic specification is particularly appropriate for \_\_\_\_\_ interface specification (Chap10).
21. Algebraic specification involves specifying \_\_\_\_\_ on an abstract data type or object in terms of their inter-relationships (Chap10).

22. An algebraic specification has a signature part defining \_\_\_\_\_ and an axioms part defining \_\_\_\_\_ (Chap10).
23. Formal specifications should have an associated \_\_\_\_\_ description to make them more readable (Chap10).
24. Algebraic specifications may be defined by defining the semantics of each inspection operation for each \_\_\_\_\_ operation (Chap10).
25. Specification should be developed incrementally from \_\_\_\_\_ specification building blocks (Chap10).
26. 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 (Chap10).
27. Given the 4 parts of a algebraic spec. structure, what does each part contribute (Chap10)?
- (a) Introduction
  - (b) Informal description
  - (c) Signature
  - (d) Axioms
28. True/False In Model-based specification methods the state of the system is not hidden unlike algebraic specification methods (Chap10/11).
29. Name the two most widely used model-based specification languages (Chap11).

- 30. Z schema include three parts (Chap11):
  - a) A \_\_\_\_\_ identifying the schema.
  - b) A \_\_\_\_\_ introducing entities and their types.
  - c) A \_\_\_\_\_ part defining invariants over these entities.
  
- 31. Z schema can be included in other schema's and may act as \_\_\_\_\_ definitions (Chap11).
  
- 32. Specification using functions does not allow ordering to be specified and sequences are used for specifying ordered collections. A sequence is a mapping from \_\_\_\_\_ integers to associated values (Chap11).
  
- 33. Operations may be specified in Z by defining their effect on the system \_\_\_\_\_ (Chap11).
  
- 34. Specifications can be used as building blocks. Specifications should be structured into small, comprehensible which are put together to form more \_\_\_\_\_ specifications (Chap11).
  
- 35. A Z specification is presented as a number of \_\_\_\_\_ , which may be combined to make new \_\_\_\_\_ (Chap11).
  
- 36. Describe what the following schema is saying (read the specification to yourself and write down the words that are expressed in the predicate) (Chap11).

```

Storage Tank
-----
contents: ℕ
capacity: ℕ
reading: ℕ
danger_level: ℕ
light: {off, on}
-----
contents    capacity
light = on <=> reading    danger_level
reading = contents
capacity = 5000
danger_level = 50
    
```

- 37. Define the six Z conventions (Chap11):



39. 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 (Chap11).

Storage Tank <span style="float: right;">(A)</span>																	
content s: $\mathbb{N}$																	
capacity: $\mathbb{N}$																	
reading: $\mathbb{N}$	(B)																
danger_level: $\mathbb{N}$																	
light: {off, on}																	
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">content s</td> <td style="padding: 2px 5px;">capacity</td> <td></td> </tr> <tr> <td style="padding: 2px 5px;">light = on</td> <td style="padding: 2px 5px;">=&lt;=&gt;</td> <td style="padding: 2px 5px;">reading</td> </tr> <tr> <td style="padding: 2px 5px;">reading =</td> <td style="padding: 2px 5px;">content s</td> <td style="padding: 2px 5px;">danger_level</td> </tr> <tr> <td style="padding: 2px 5px;">capacity =</td> <td style="padding: 2px 5px;">5000</td> <td style="padding: 2px 5px;">(C)</td> </tr> <tr> <td style="padding: 2px 5px;">danger_level =</td> <td style="padding: 2px 5px;">50</td> <td></td> </tr> </table>			content s	capacity		light = on	=<=>	reading	reading =	content s	danger_level	capacity =	5000	(C)	danger_level =	50	
content s	capacity																
light = on	=<=>	reading															
reading =	content s	danger_level															
capacity =	5000	(C)															
danger_level =	50																

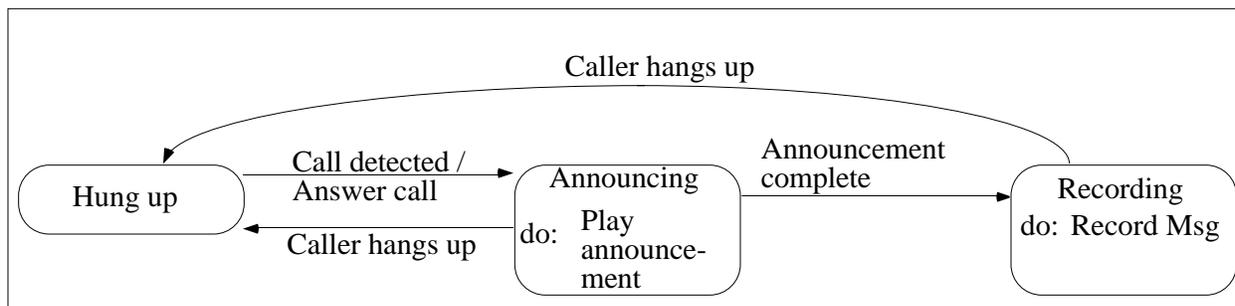
- A. \_\_\_\_\_
- B. \_\_\_\_\_
- C. \_\_\_\_\_
- D. \_\_\_\_\_
- E. \_\_\_\_\_
- F. \_\_\_\_\_
- G. \_\_\_\_\_

ARRAY (Elem: [Undefined -> Elem])	(D)
<b>sort</b> Array	
<b>imports</b> INTEGER	
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 element assigned the given value. Eval reveals the value of a specified element.	(E)
If an attempt is made to access a value outside the bounds of the array, the value is undefined.	
Create (Integer, Integer) -> Array	
Assign (Array, Integer, Elem) -> Array	(F)
First (Array) -> Integer	
Last (Array) -> Integer	
Eval (Array, Integer) -> Elem	
First (Create (x, y)) = x	
First (Assign (a, n, v)) = First (a)	
Last (Create (x, y)) = y	
Last (Assign (a, n, v)) = Last (a)	(G)
Eval (Create (x, y), n) = Undefined	
Eval (Assign (a, n, v), m) =	
<b>if</b> m < First(a) <b>or</b> m > Last(a) <b>then</b>	
Undefined <b>else</b>	
<b>if</b> m = n <b>then</b> v <b>else</b> Eval (a, m)	

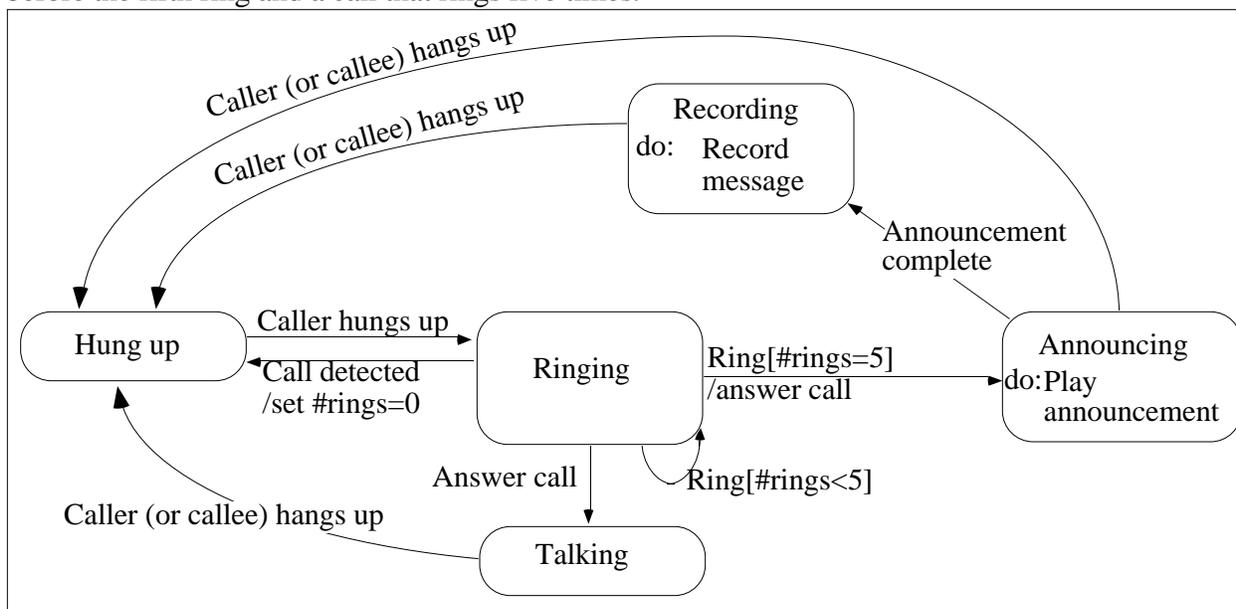
40. [1pts] Specifications should be developed incrementally from simpler specification \_\_\_\_\_ (Chap11).

41. [1pt] Formal specification forces an analysis of the system requirements at an \_\_\_\_\_ (Chap11).

42. [19pts] Consider the Telephone Answering Machine (TAM) shown in the diagram below (Chap11).



The TAM above is activated on the first ring. Revise the state diagram so that the TAM answers after five rings. However, if the telephone is answered before five rings, the machine should do nothing (except to allow Talking). Be careful to distinguish between calls where the callee answers before the fifth ring and a call that rings five times.



The process of writing a formal specification based on design artifacts may be called \_\_\_\_\_ [1pt].

Choose one of the two types of formal specification methods that we looked at in class (Chapter 10 and 11) and in the study questions and quizzes and give your best answer for the TAM problem above.

Note, there is definitely one of the two methods more suited for this particular problem and points will be given based on that decision, points will be awarded on the completeness of your answer, points will be given on the syntax (or grammar). The rest of the points (6) will be given based on a comparison to other answers in conjunction with how elegant (including NEATNESS) your answer is.

Place your answer below:

