

SAMPLE EXAM

SOFTWARE CONSTRUCTION AND USER INTERFACE

COM S/ SE 319

Iowa State University
Department of Computer Science

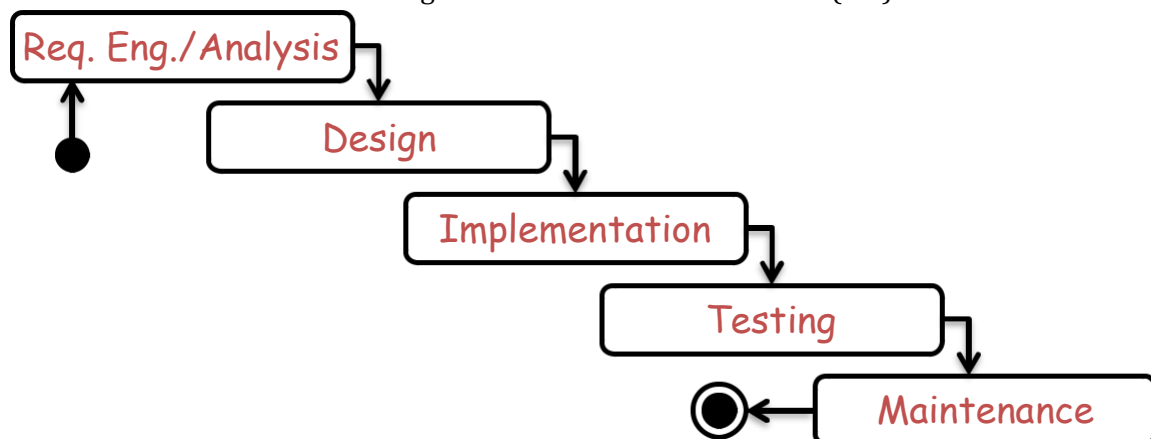
Task	1	2	3	4	5	Σ
Maximum	15	30	20	10	25	100

TASK 1: WARM-UP (15 POINTS)

- a) True or false? Tick whether the statement is true or false. Each right cross gives a point, a point is deducted for each false cross. The task is scored at least 0 point (no negative points). (5 P)

Statement	True	False
Non-functional requirements are not part of the specifications.		✗
Control flow-oriented tests and data-flow-oriented tests belong to the static analysis of programs.		✗
A composition is an aggregation in which the parts do not exist without the whole.	✗	
TDD can be used in almost every development phase to detect bugs at late developmental stages.		✗
Contravariant input parameters fulfill the substitution principle.	✗	

- b) Complete the following diagram of the waterfall model by entering the names of the phases of the waterfall model in the given boxes in the correct order. (5 P)



- c) Define the terms "signature inheritance" and "implementation inheritance" from object-oriented programming. How is the relationship between them? (3P)

Signature inheritance: A defined and (possibly) implemented method in the superclass only transfers its signature to the subclass. (1P)

Implementation inheritance: A defined and implemented method in the superclass transfers its signature and implementation to the subclass. (1P)

Implementation inheritance does not work without signature inheritance, but conversely it works. (1P)

- d) Which points must be fulfilled so that we can speak of a structure-preserving transformation? Complete the following definition. (2 P)

We speak of a structure-preserving transformation of a source language into the intermediate language, if ...

Per point: 1 P

(Exclusively) the commands affecting the execution order are replaced by intermediate language command sequences, where the order of execution of the other commands remains the same with the same parameterization with that in the source language! (1P)

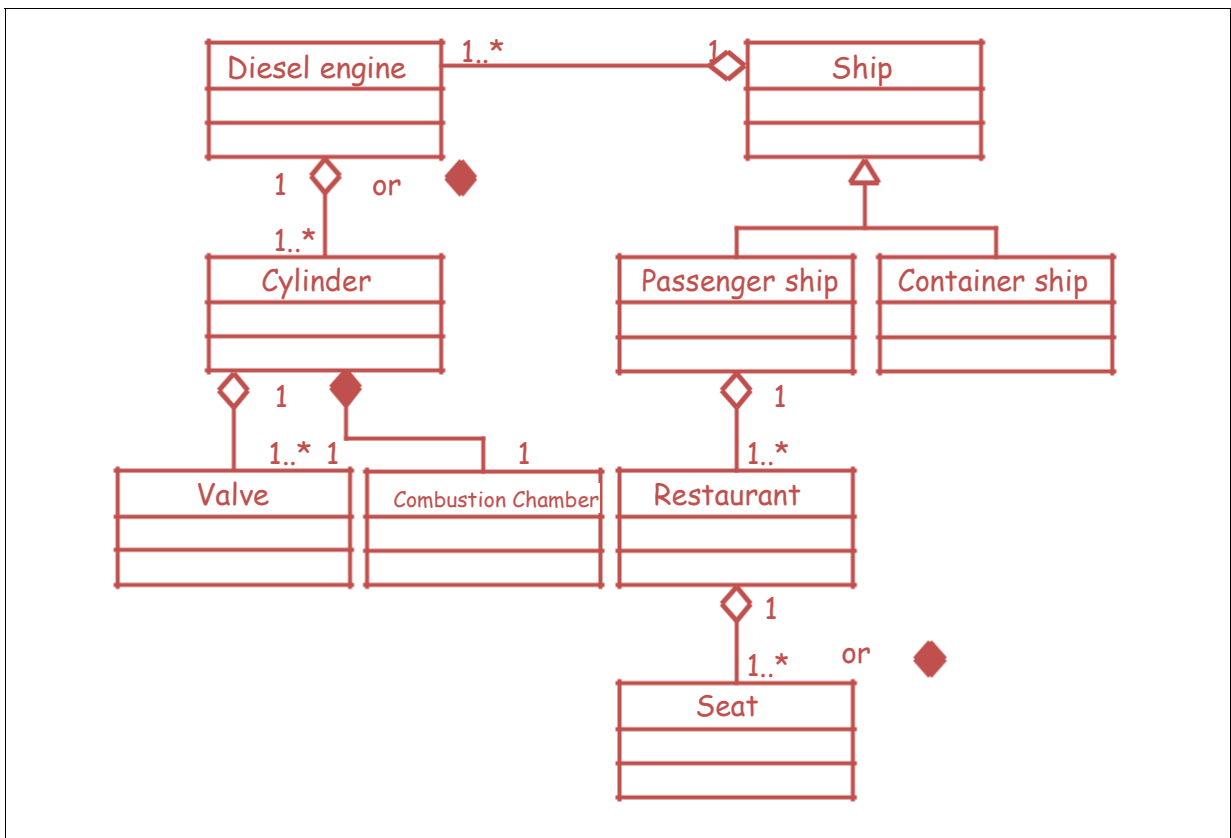
all other commands are taken over unchanged (1P)

TASK 2: SYSTEM MODELING AND UML DIAGRAMS (30 POINTS)

- a) Consider the following scenario:

Ships are either passenger or container ships. A passenger ship may have one or more restaurants, each providing a certain number of seats. Every ship has at least one diesel engine. A diesel engine consists of several cylinders. Each cylinder has valves and a combustion chamber.

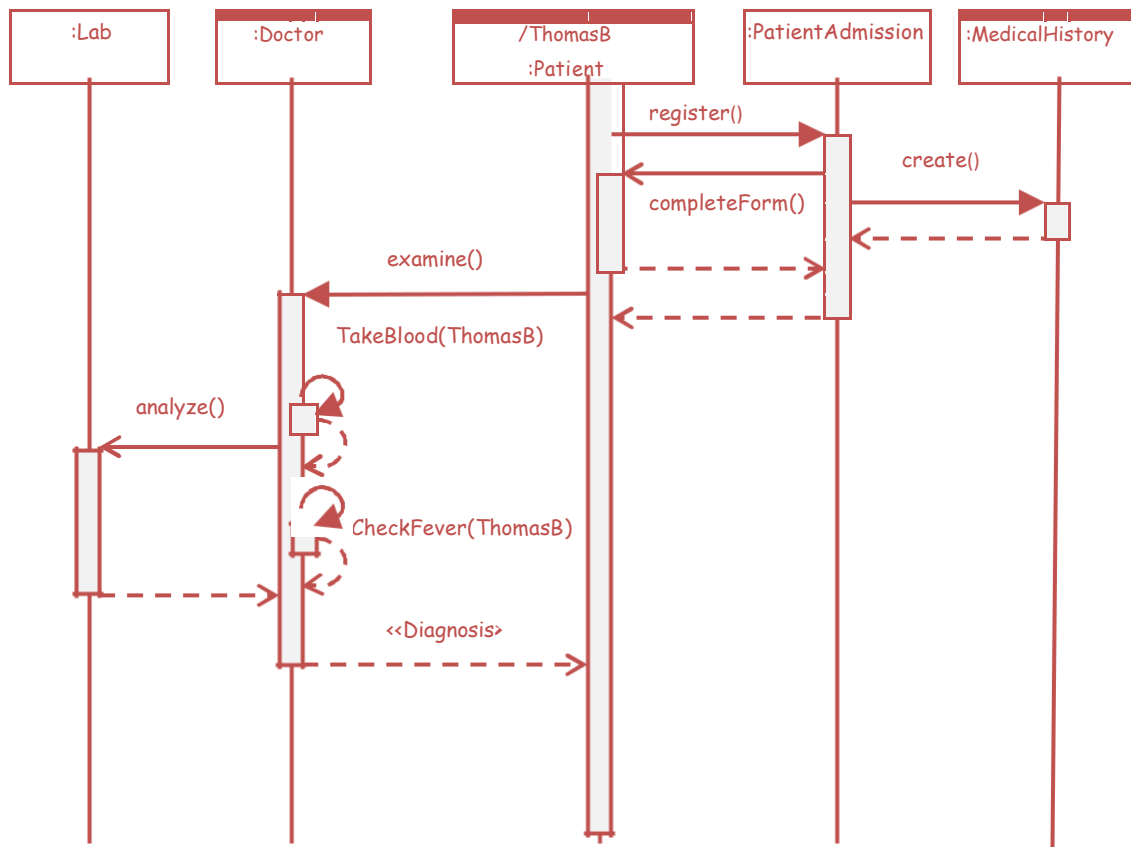
Use UML class diagram to model the above scenario (as completely as possible). (15 points)



Ship-relation (inheritance): 1 P
 each class (except 2 given ones): $7 \times 1 = 7P$
 Aggregation: $6 \times 1 = 6P$
 Cardinality: 1P

- b) Draw a sequence diagram with appropriate messages, attributes, activation bar (lifeline, objects are given) for the following scenario. (15 points)

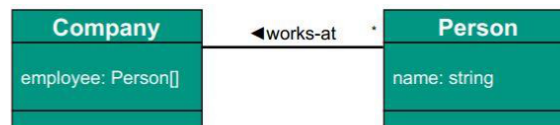
Thomas B. does not feel well and wants to be examined by a doctor to get a diagnosis. He goes to the hospital and registers on to the patient admission. While the clerk creates the medical record at the patient admission, Thomas B. completes the medical history form given to him by the clerk. After Thomas B. has returned the completed form to the clerk, Thomas B. is examined by the doctor. For this, first the doctor takes blood sample from Thomas B. While the lab analyzes the blood, the doctor checks his fever. Finally, Thomas B. gets his diagnosis from the doctor.



For Message/ Function: $6 \times 1 = 6P$
 For Solid/Dashed Line: $6 \times 1 = 6P$
 For Activation Bar: $3 \times 1 = 3P$

TASK 3: XP, OBJECT MODELING, AND... (20 POINTS)

a) Is the following model meaningful? Explain your reasons? (4P)



No, It is bad because employee is modeled both by association and by attribute. The employee array is just the implementation of the association!

b) When do we use Spikes in XP process model? (4P)

When we want to do experimental work /investigations. We have not done before.

"Never made anything comparable"

Doing experiment.

Once through the whole problem.

Do not aim for a perfect solution.

c) Describe three differences between an interpreter and a compiler? (6P)

Each 2p

1) Interpreter generates intermediate object code and is memory efficient.

2) Interpreter needs less amount of time to analyze the source code but the overall execution time is slower.

3) debugging with interpreter is easier compare with compiler

d) Define white-box testing and black-box testing? (4P)

White-Box test cases determine the test values with knowledge of control and / or data flow. (2p)

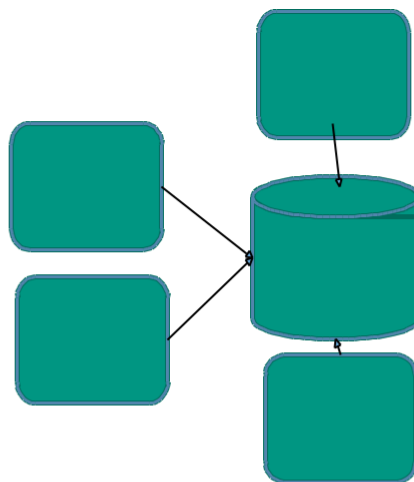
Black-Box test cases determining the test values without knowledge of control and / or data flow; just out of specification. (2P)

e) Why a concurrent automaton (state diagram) is not more powerful than a "normal" finite automaton? (2p)

because you can simulate the parallelism with eventually many additional states (2p).

TASK 4: ARCHITECTURAL STYLE (10 POINTS)

a) Which architectural style is used in the following figure? What are the advantages and disadvantage for this architecture? (6P)



Data-centric, (repository) (2p)

Weak coupling across its components (2p) single-point-of-failure or performance bottleneck in this architecture style (2p)

- b) Explain a layered architecture. Is a client/server a layered architecture? (4P)

Layered architecture: A layer only uses services of lower layers and does not use higher layers (2p), yes (2P)

TASK 5: TESTING (25 POINTS)

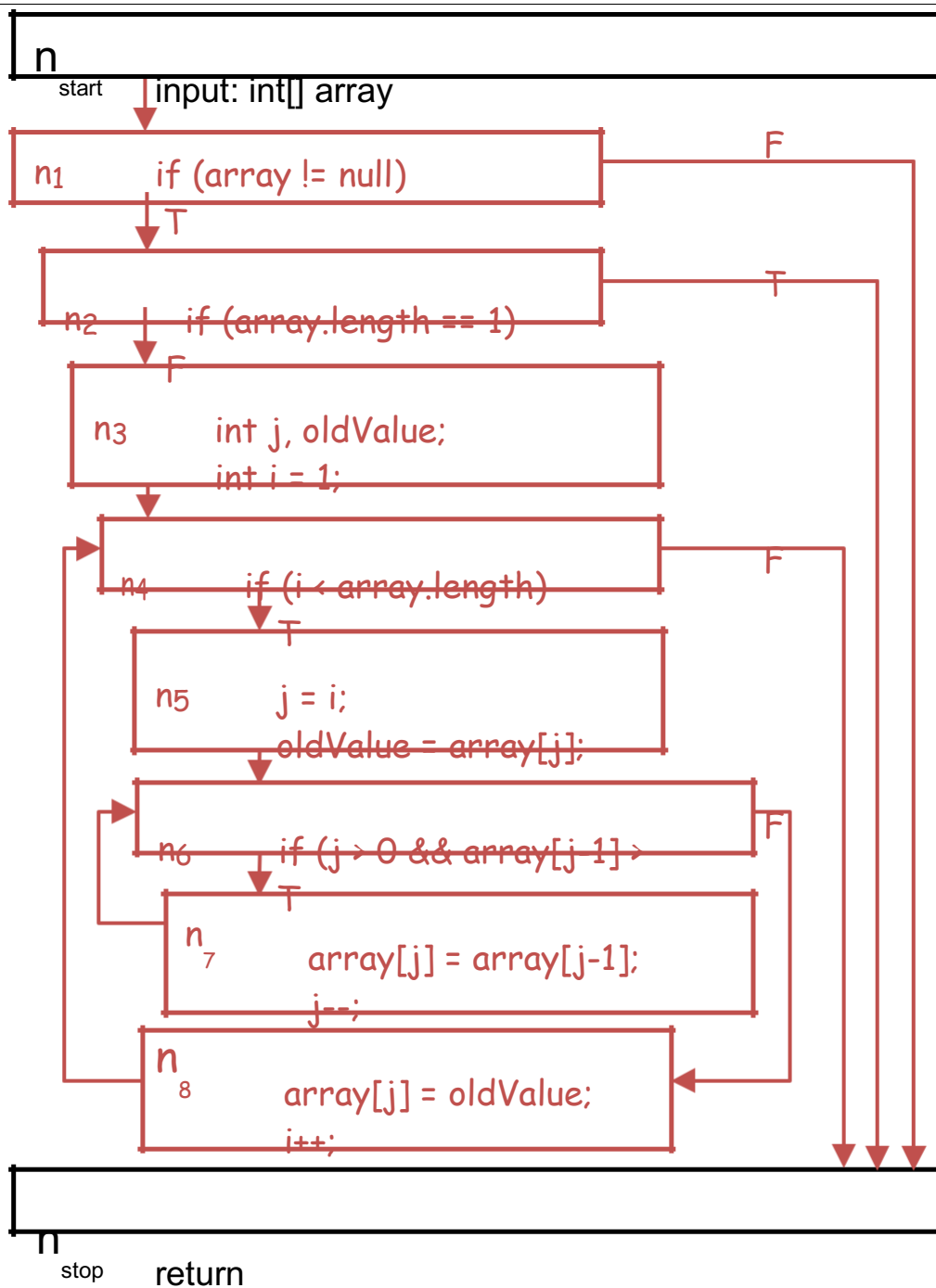
Consider the following sort(...) method, which sorts an array of integers in ascending order (sorting by insertion):

```
1 public void sort(int[] array) {
2     if (array != null) {
3         if (array.length == 1) {
4             return;
5         } else {
6             int j, oldValue;
7             for (int i = 1; i < array.length; i++) {
8                 j = i;
9                 oldValue = array[i];
10                while (j > 0 && array[j - 1] > oldValue) {
11                    array[j] = array[j - 1];
12                    j--;
13                }
14                array[j] = oldValue;
15            }
16        }
17    }
18 }
```

- a) In the given box on the next page, create the control flow graph of the sort(...) method. (20 P)
- b) Specify a minimum test case set (minimum set of input numbers) that satisfies the statement coverage for the sort(...) method. Enter the paths that have been traversed. (5 P)

{2, 1}: nstart, n1, n2, n3, n4, n5, n6, n7, n6, n8, n4, nstop

5 P



each if-condition: 3 P (4 if × 3p = 12p)
 Rest: 8 P (4 BB × 2p = 8p)