
2910226 Software engineering, algorithm design and analysis

Examination paper: Zone A

Time allowed: three hours

Full marks will be awarded for complete answers to a total of **four** questions.

You must answer **two** questions from Section A and **two** questions from Section B.

Each question carries 25 marks. The marks for each part of a question are indicated at the end of the part in [] brackets.

There are 100 marks on the paper.

No calculators may be used.

Section A

Question 1

(a) Define briefly what the following terms mean in a software engineering context:

- (i) module
- (ii) encapsulation
- (iii) coupling
- (iv) interface
- (v) context dependencies
- (vi) abstraction
- (vii) cohesion

[14 marks, 2 marks for each term]

(b) What is a good software system like? That is, what structural characteristics do good software systems typically have? Be sure to use the correct software engineering terms in your answer.

[10 marks]

(c) State whether the following statement is true or false: "A module may have two or more interfaces".

[1 mark]

Question 2

(a) Draw a UML class diagram which shows two classes, A and B. Indicate on the diagram that each instance of class A can be associated with 1 or more instances of class B and that each instance of class B can be associated with up to 2 instances of class A. Indicate on your diagram that messages can be sent from A to B but *not* from B to A.

[6 marks]

(b) Could the association between the classes A and B be an aggregation? If so, redraw the diagram in part (a) to represent this, omitting all other indications of navigability. If the association could not be an aggregation, explain why.

[3 marks]

(c) Could the association between the classes A and B be a composition? If so, redraw the diagram in part (a) to represent this, omitting all other indications of navigability. If the association could not be a composition, explain why.

[3 marks]

(d) Draw a UML class diagram which shows an association between two classes, C and D. Indicate on the diagram that the nature of the association is that D *attacks* C. Indicate on the diagram that, in this association, instances of D take on the role of "aggressor" and instances of C take on the role of "victim". Indicate on your diagram that an association class is used to store information about the weapon used, the time, the date and the location of each attack.

[6 marks]

(e) Draw a UML class diagram that represents the same information as that drawn in part (d) but, in addition, allows us to represent that a particular aggressor attacked a particular victim on more than one occasion. Place appropriate multiplicities on your new diagram, assuming that a given victim could be attacked by one or more aggressors and that a given aggressor could attack 1 or more victims.

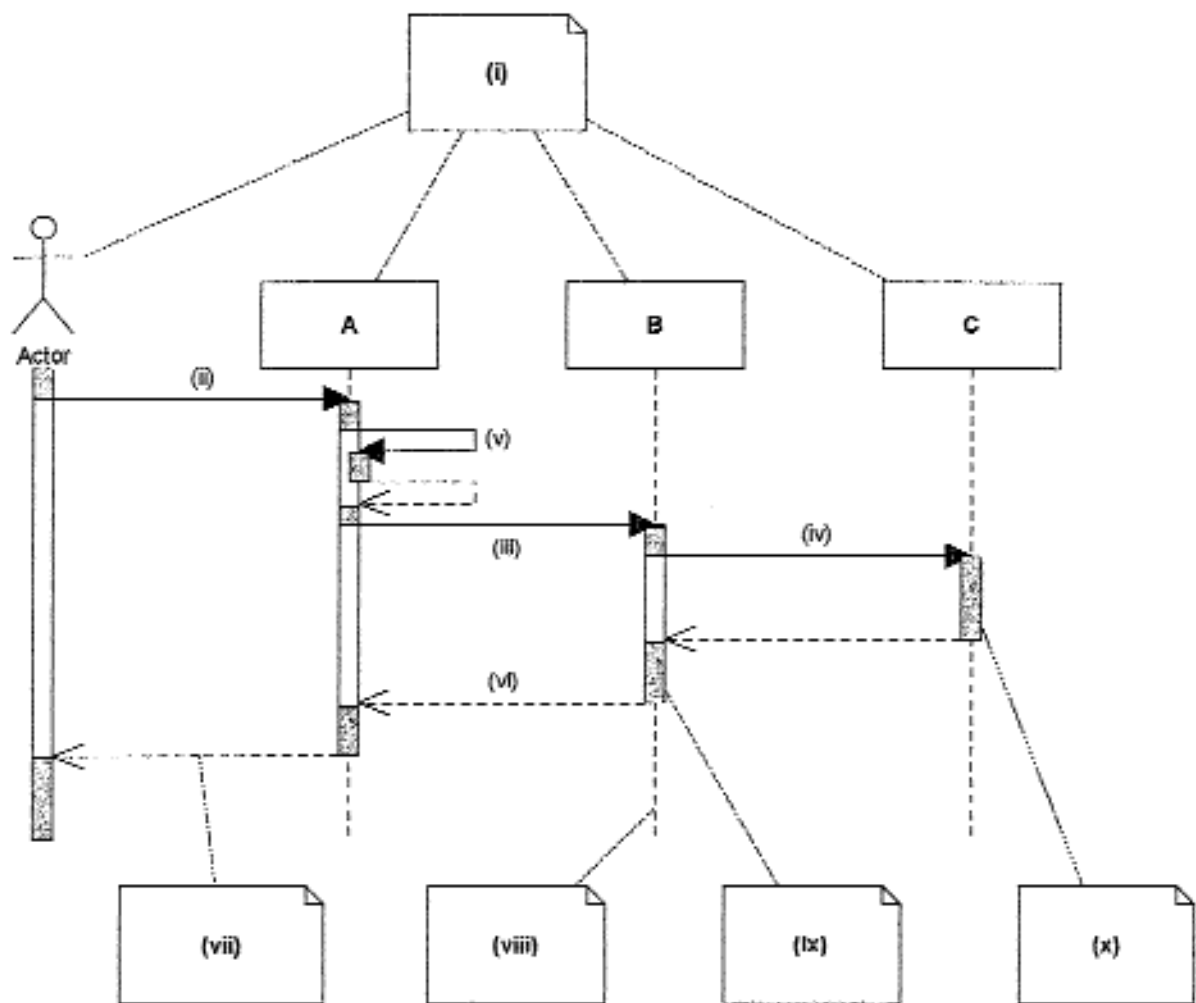
[5 marks]

(f) Explain the difference between a *class model* and a *class diagram*.

[2 marks]

Question 3

Consider the UML diagram shown below and answer the questions that follow it.



- (a) What kind of diagram is this and what are such diagrams usually used to represent? [3 marks]
- (b) Does this diagram represent the structure or the behaviour of a system? [1 mark]
- (c) What are the elements labelled (i)? [1 mark]
- (d) What do the arrows labelled (ii), (iii) and (iv) represent? [1 mark]
- (e) Which happens first: (ii), (iii) or (iv)? [1 mark]
- (f) Explain the difference between a synchronous and an asynchronous message. [4 marks]
- (g) Does this diagram contain synchronous or asynchronous messages? [1 mark]
- (h) What does the arrow labelled (vii) represent? [1 mark]
- (i) What does the dashed line labelled (viii) represent? [1 mark]
- (j) What do the thin vertical rectangles labelled (ix) and (x) represent? [1 mark]
- (k) Does the diagram represent a concurrent or a procedural interaction? [1 mark]

(Question 3 continued)

- (l) If (vi) were replaced with a label, what would that label represent? [1 mark]
- (m) What does the arrow labelled (v) represent and what would this correspond to if A was an object in an object-oriented system? [3 marks]
- (n) What does the grey shading represent on the thin vertical rectangle labelled (ix)? Why is only part of the rectangle shaded? [3 marks]
- (o) If A and C represent objects, is there necessarily an association in the class model for this system between the classes to which A and C belong? Explain your answer. [2 marks]

Section B**Question 4**

- (a) Distinguish, with the aid of examples, the concept of a *problem* and an *instance* of a problem. [5]
- (b) Let X be an array of n elements. Algorithm A executes an $O(n)$ time computation for each even number in X , and an $O(\log n)$ time computation for each odd number in X . What is the worst time and the best time complexity for A ? [5]
- (c) Analyse the following algorithm. Suppose that the array A contains, initially, 4, 5, 6, 4, 3, 2, 2, 2. What are the elements in A on completion of the execution of the algorithm? Hence state what the purpose of the algorithm might be and how well it suits the purpose. [5]

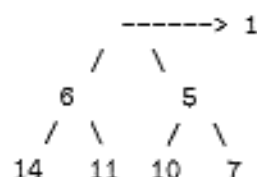
```

1: for  $i = 1, i \leq n, i \leftarrow i + 1$  do
2:    $j \leftarrow i + 1$ 
3:   while  $j < n$  do
4:     if  $A[i] = A[j]$  then
5:        $A[j] \leftarrow -A[i]$ 
6:        $j \leftarrow j + 1$ 
7:     end if
8:   end while
9: end for

```

- (d) Consider an ADT list of integers. Write a *typed* method `listTotal()` that computes the sum of the integers in the list. Assume the list is reference based. [5]
- (e) Explain, with the example below, what is meant by a *min-heap* and how the order property of the heap is maintained after removing the root node '1'. Show all the steps. [5]

Binary heap:



Question 5

- (a) Describe in plain English three operations of an ADT two dimensional table. You should include necessary parameters and other details. [5]
- (b) Explain, with the aid of an example, what is meant by the term *collision* in the context of hashing. [5]
- (c) Suppose we want to store data (3,4,5,26,6,7,23,16,39,17,22,55) in the hash table $\Pi[0..22]$ and use the hash function $h(k) = k \bmod 23$. Demonstrate the content of the hash table using the linked list closed addressing. Is there any need for rehashing? How many times would you attempt (the probing) before being successful? [5]
- (d) Write the execution result of the following program. [5]

```
class Syracuse {
    static void syr(int n) {
        System.out.println(n);
        if (n>1) {
            if (n%2==0) syr(n/2);
            else syr(3*n+1);
        }
    }

    public static void main(String[] args) {
        syr(12);
    }
}
```

- (e) Would it be possible to construct a binary search tree of the *minimum height* for the integers 'e' 'g' 'b' 'j' 'h' 'k' 'd' 'c' 'f' 'a'? If yes, draw such a tree and rewrite the given data in order that the binary search tree constructed. If no, justify your conclusion with an example. [5]

Question 6

- (a) Draw a single binary tree *T* to store an arithmetic expression such that [5]

- (i) Each node of *T* stores a single operand or operator.
- (ii) A postorder traversal of *T* yields *DEA - *FBC + /+*.

- (b) Describe, in one sentence, the main operational characteristics of a queue. What would be the content of the initially empty queues *queueA* and *queueB* after accomplishment of the following sequence of operations? [5]

```

queueA.enqueue(1)
queueA.enqueue(2)
queueB.enqueue(3)
queueB.enqueue(4)
tmp=queueA.dequeue()
tmp=queueB.peak()
queueA.enqueue(tmp)
queueA.enqueue(5)
tmp=queueB.dequeue()
queueB.enqueue(6)

```

- (c) A stable sorting algorithm maintains the relative order of repeated elements. Using an array of integers (5 7 6 8 5 9 1 2) as an example, demonstrate the stability of the *insertion sort* algorithm. [5]

- (d) Define each of the complexity classes below. Explain briefly how to prove a new problem is NP-complete. [5]

- (i) NP
- (ii) NP-complete

- (e) Explain, with the aid of an example below, what a **compressed trie** is. Demonstrate how the following words are stored in the data structure. [5]

(predate, precise, precinct, poor, inject, inner, inroad, key, kettle).