2910226 Software engineering, algorithm design and analysis

Examination paper: Zone B

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) "An object is a *thing* which has *behaviour*, *state* and *identity*." Explain and discuss what this statement means, focusing on the italicised terms.

[10 marks]

(b) Provide brief definitions for the following terms in the context of object-oriented software development. Use examples where appropriate.

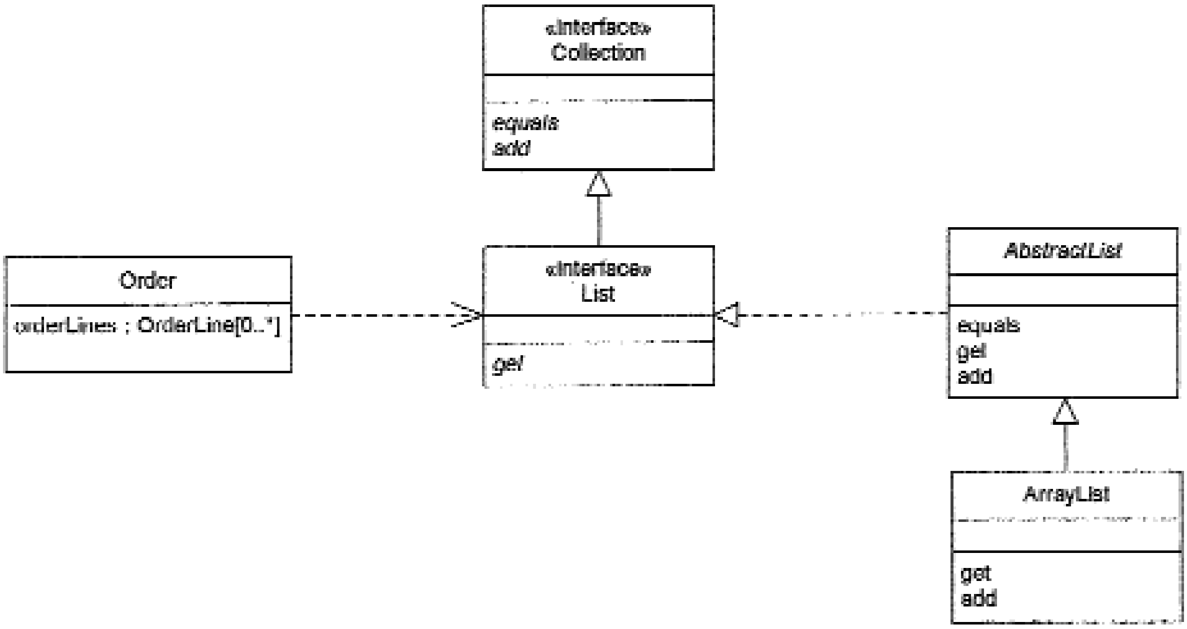
(i) method selector [2 marks]

{ii} class definition [3 marks]

(iii) instantiating a class [1 mark]

[6 marks for part (b)]

(c) Consider the following diagram and answer the questions that follow it.



(i) What kind of diagram is this? [1 mark]

(ii) What does the arrow from the box labelled "List" to the box labelled "Collection" signify? [1 mark]

(iii) What does the arrow from the box labelled "Order" to the box labelled "List" signify? [1 mark]

(iv) What does the arrow from the box labelled "*AbstractList*" to the box labelled "List" signify? [1 mark]

(v) Can AbstractList be instantiated? Explain your answer. [2 marks]

(vi) What methods must be implemented in order for a class to implement the List interface? [3 marks]

[9 marks for part (c)]

Question 2

(a) Name the three modes of use for UML and briefly describe the main feature of each.

[6 marks]

(b) For each of the following types of UML diagram, identify whether it is a Structure diagram or a Behaviour diagram:

(i) Class diagram

(ii) Use case diagram

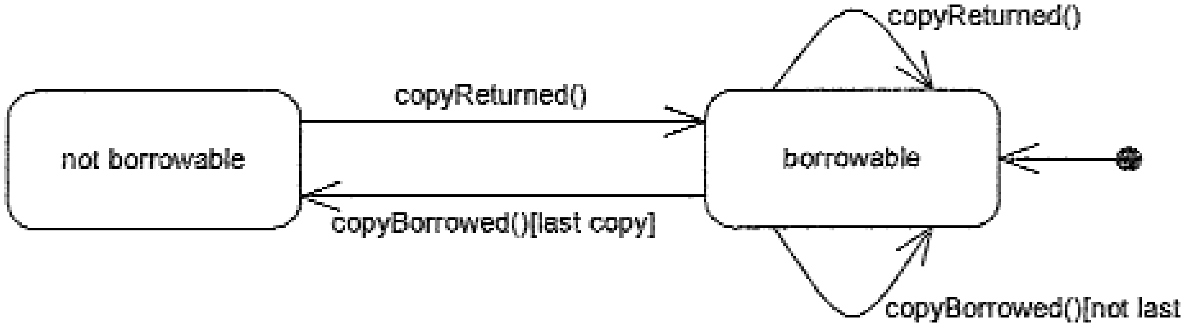
(iii) Package diagram

(iv) State machine diagram

(v) Sequence diagram

[1 mark each, 5 marks for part (b)]

(c) Consider the following state machine diagram which shows how the state of a Book object in a library software system changes between "borrowable" and "not borrowable".



(i) According to this diagram, what state does a Book object start off in, "borrowable" or "not borrowable"? [1 mark]

(ii) Under what conditions does the state of a Book object change from being "borrowable" to "not borrowable"? [2 marks]

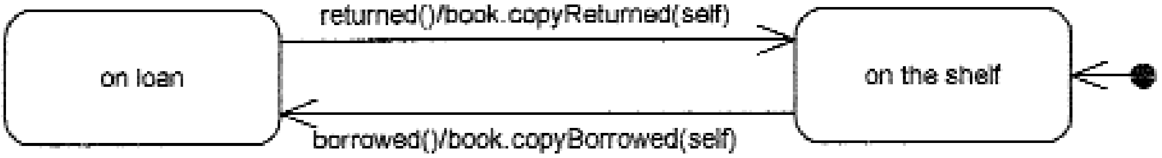
[3 marks for part (c)]

(Question 2 continued overleaf)

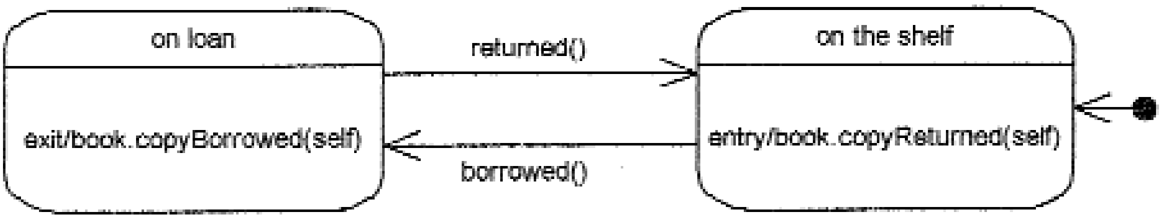
(Question 2 continued)

(d) Consider the following diagrams which relate to a Copy object in a library software system and answer the questions that follow.

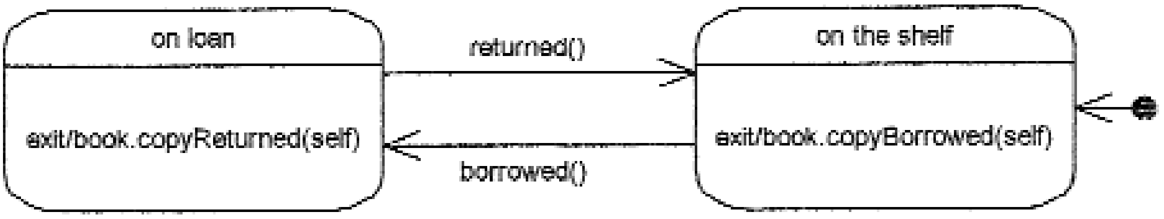
(1)


on loan
returned()/book.copyReturned(self)
on the shelf
borrowed()/book.copyBorrowed(self)

(2)



(3)



(i) Which two of the three diagrams, (1), (2) and (3), are equivalent? [2 marks]

(ii) In diagram (2), under what conditions is the `book.copyBorrowed(self)` message sent? [2 marks]

(iii) To what does the argument "self" refer? [1 mark]

[5 marks for part (d)]

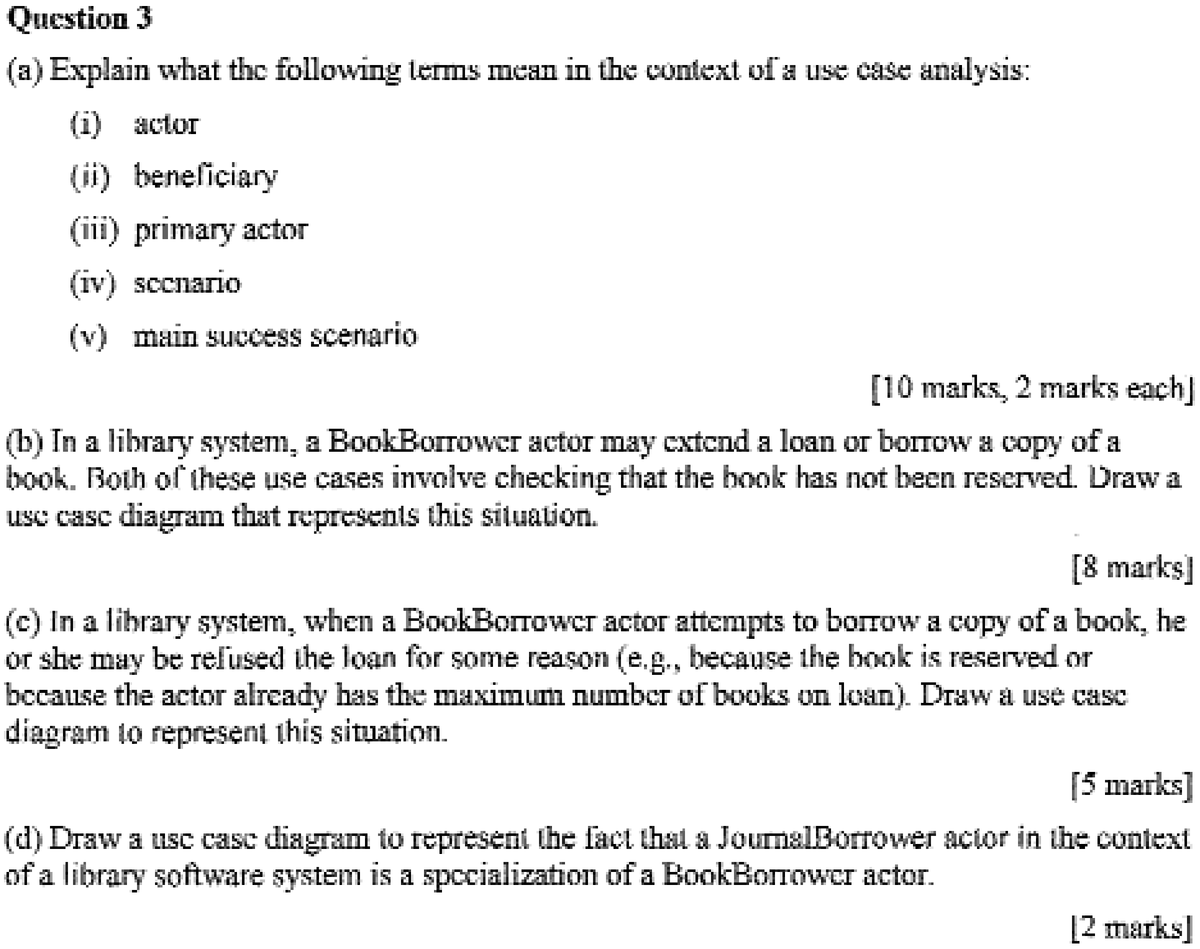
(e) A hardware update wizard can be in three states as follows:

1. Displaying a hardware update window.
2. Searching for new hardware.
3. Displaying new hardware found.

The wizard starts by displaying a hardware update window. While displaying this window, the user can press a "Search" button to cause the wizard to start searching for new hardware, or the user can press a "Finish" button to leave the wizard. While the wizard is searching for new hardware, the user may cancel the search at any time. If the user cancels the search, the wizard displays the hardware update window again. When the wizard has completed searching for new hardware, it displays the new hardware found.

Draw a state machine diagram that represents the function of the hardware update wizard just described.

[6 marks]



Question 3

(a) Explain what the following terms mean in the context of a use case analysis:

(i) actor

(ii) beneficiary

(iii) primary actor

(iv) scenario

(v) main success scenario

[10 marks, 2 marks each]

(b) In a library system, a BookBorrower actor may extend a loan or borrow a copy of a book. Both of these use cases involve checking that the book has not been reserved. Draw a use case diagram that represents this situation.

[8 marks]

(c) In a library system, when a BookBorrower actor attempts to borrow a copy of a book, he or she may be refused the loan for some reason (e.g., because the book is reserved or because the actor already has the maximum number of books on loan). Draw a use case diagram to represent this situation.

[5 marks]

(d) Draw a use case diagram to represent the fact that a JournalBorrower actor in the context of a library software system is a specialization of a BookBorrower actor.

[2 marks]

Section B

Question 4

(a) The terms *abstract data structure* and *abstract data type* are often used interchangeably. Explain briefly, with an example, when *abstract data structure* is more appropriate and when *abstract data type* is appropriate to use. [5]

(b) Let X be an array of n elements. Algorithm A chooses $\log n$ elements in X at random and executes an $O(n)$ time calculation for each. Outline algorithm A in pseudocode and compute the worst-case running time of Algorithm A step by step. [5]

(c) Consider the following method. 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]

```
for $i = 1, i \leq n, i \leftarrow i + 1$ do
    $j \leftarrow i + 1$
    while $j < n$ do
        if $A[i] = A[j]$ then
            $A[i] \leftarrow A[i]$
            $j \leftarrow j + 1$
        end if
    end while
end for
```

(d) Write a pseudocode method `swap(aList, i, j)` that interchanges the items currently in positions i and j of a list. Define the method in terms of the operations of the ADT list such as `getItem(i)`, and `removeItem(j)`. [5]

Hint: The method should be independent of any particular implementation of the list. You may define any new interfaces if necessary, and include comments to clarify your ideas.

(e) 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 preorder traversal of T yields $+ * D - EA/F + BC$.

Question 5

(a) Explain with an example what dynamic programming is and when it is particularly useful. [5]

(b) What is so-called "greedy approach"? Describe, with an example, a computational problem which can be solved by a greedy approach? [5]

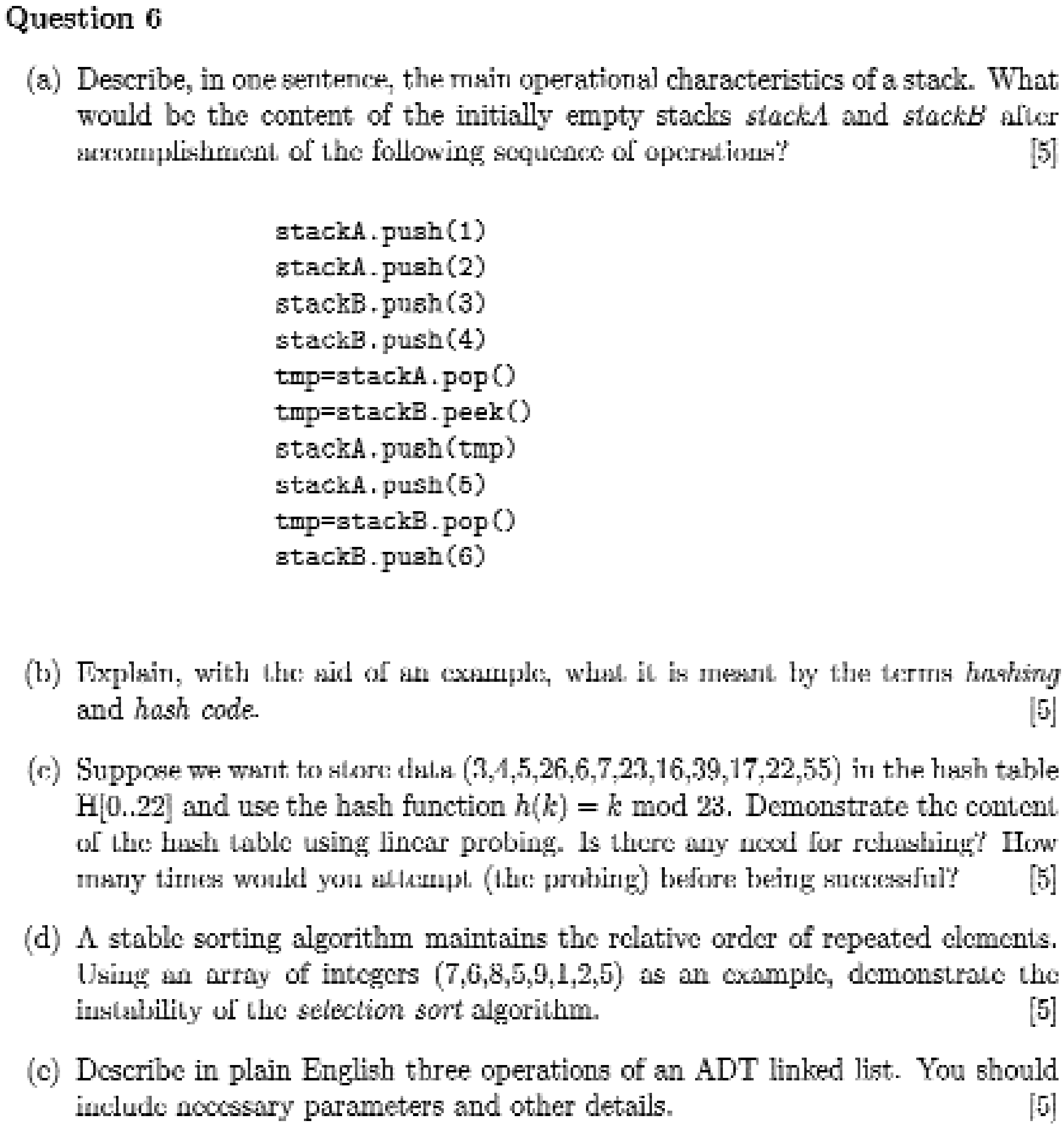
(c) Write the execution result of the following program. [5]

```
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(13);
        }
}
```

(d) Would it be possible to construct a binary search tree of the *minimum height* for integers 5 7 2 9 8 10 4 3 6 1? 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]

(e) Explain, with the aid of an example, what is meant by the word *problem* and what it is meant by *solution* in context of Algorithm Design. [5]



Question 6

(a) Describe, in one sentence, the main operational characteristics of a stack. What would be the content of the initially empty stacks *stackA* and *stackB* after accomplishment of the following sequence of operations? [5]

```
stackA.push(1)
stackA.push(2)
stackB.push(3)
stackB.push(4)
tmp=stackA.pop()
tmp=stackB.peek()
stackA.push(tmp)
stackA.push(5)
tmp=stackB.pop()
stackB.push(6)
```

(b) Explain, with the aid of an example, what it is meant by the terms *hashing* and *hash code*. [5]

(c) Suppose we want to store data (3,4,5,26,6,7,23,16,39,17,22,55) in the hash table H[0..22] and use the hash function $h(k) = k \bmod 23$. Demonstrate the content of the hash table using linear probing. Is there any need for rehashing? How many times would you attempt (the probing) before being successful? [5]

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

(c) Describe in plain English three operations of an ADT linked list. You should include necessary parameters and other details. [5]