NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

EXAMINATION FOR

Semester 2 : 2000/2001

CS2103/CS2103S/TDCS2103 – SOFTWARE ENGINEERING

April/May 2001 Time Allowed 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **FIVE(5)** questions and comprises **NINE(9)** printed pages.
2. Answer **ALL** questions.
3. Answer **ALL** questions within the space provided in this booklet.
4. This is an **Open Book** examination.
5. You may use the back blank pages.
6. You may write in either pencil or ink. Please write legibly!
7. **Please write your Matriculation Number Below.**

MATRICULATION NO: ____________________

This portion is for examiner's use only

Question	Marks	Remark
Q1		
Q2		
Q3		
Q4		
Q5		
Total		

Question 1 (30 marks)

You are to design a simple XYZ Airline Portal System. This system maintains information about flights operated by the XYZ Airline. Each flight is associated with an origin, a destination, and optionally a transit airport. For example, XY001 flies from Singapore to London via Bangkok airport. In addition, it maintains departure/arrival timing information, e.g. XY001 departs Singapore at 0800 hours, arrives Bangkok 1000hrs, departs Bangkok 1100hrs and finally arrives London 1700hrs. Every flight flies daily, and there could be multiple flights bound for the same location.

There is a maximum number of passengers for each class on each flight. For example, XY006 can take 10 First Class, 20 Business Class and 150 Economy Class passengers. The system also maintains the current number of bookings for each flight on each day.

A user may make **enquiries** on the system. For simplicity, we assume that the only enquiry is on the timing of a given flight and airport, e.g. "*What is the arrival time for flight XY001 into London*?".

A user may **request a booking** by specifying the origin, destination, departure date, return date, and the flight class. For simplicity, assume no one-way trip, no stopover, and passenger must return from the same destination airport. If there exist flights, both departure and return, on the specified dates and class that are not yet full, the system will proceed to confirm the booking and update the system. Otherwise, the system will reject the booking request. Note that the origin or destination may be a flight transit airport. For simplicity, assume that the user cannot choose nor reject flights assigned by the system.

Based on the above specification:

(a) design a class diagram (which should contain classes, attributes and relationships) (14 marks)
(b) design a collaboration diagram for the **enquiry** scenario stated in *italics* above (6 marks)
(c) design a sequence diagram for the **request a booking** scenario (10 marks)

(a) Class diagram:

(b) Collaboration diagram:

(c) Sequence diagram:

Question 2 (20 marks) [2 marks for each correct answer, 0 for a wrong answer or no answer]

Circle your answer.

1. If class A is a subtype of class B, then all code that operates correctly on instances of class B will operate correctly on instances of class A. **True / False**

2. In a Java program, if class A extends class B, then A is a subtype of B. **True / False**

3. A strong precondition can allow a more efficient implementation of a method. **True / False**

4. Strong preconditions tend to simplify client code. **True / False**

5. If method A is only called by B, and B ensures that A's precondition is met, then the precondition can be removed from A's specification. **True / False**

6. If `y=x.clone()` using shallow copy and `z=y.clone()` using deep copy, then `z` and `x` are equal by deep equality. **True / False**

7. Consider the following Java program fragment:

```
class A{
 ...
 public void foo() {...}
}

class B extends A{
 ...
 public void foo() {...}
}
:
b = new B();
b.foo();
```

When `b.foo()` is being called, it is possible that the `foo()` method within `A` is executed. **True / False**

8. When a module is replaced by another module with identical specification, integration testing should be repeated. **True / False**

9. In bottom-up integration testing, each module is tested after all the modules it depends on. **True / False**

10. Where data access is infrequent, we should introduce a derived association to improve the efficiency of the data access. **True / False**

Question 3 (15 marks)

(a) Two important concepts of OO programming is polymorphism and inheritance. Using one sentence each, state one advantage and one disadvantage of each concept. (4 marks)

Polymorphism advantage:____

____

Polymorphism disadvantage:______________________________________

____

Inheritance advantage:____

____

Inheritance disadvantage:____

____

(b) Consider the `Network` ADT discussed in the assignments. Suppose it is required to compute several other forms of optimal paths (besides the shortest and quickest) that are also based on costs on edges. Using polymorphism, propose a detailed design (PDL notation) for the `Network` ADT. *Hint:* You may need to define additional classes. (11 marks)

(Question 3, cont'd)

Question 4 (20 marks)

A PQ (priority queue) is an abstraction that stores objects with their associated priority, and allows the objects to be retrieved in priority order. Consider the following detailed design of PQ:

```
public class PQ{
// A priority queue is a mutable collection of object/priority pairs {<o1,p1> , ..., <on,pn>}.
// Priorities are integers between 0 and 9, inclusive.

public PQ()
// Post-Condition: makes an empty priority queue.

public void append(Object o,int p) throws IllegalArgumentException
// Post-Condition: if p < 0 or p > 9 or o = null throws IllegalArgumentException,
// else adds <o, p> to the priority queue.

public Object remove() throws IllegalArgumentException
// Post-Condition: if the priority queue is empty throws IllegalArgumentException,
// else removes one element of the highest priority from the priority queue and returns it.
}
```

Example

Consider the PQ instance pq = {<x,7>, <z,5>, <y,7>}; here x and y have priority 7 and z has priority 5. Calling pq.remove() will cause either <x,7> or <y,7> to be removed from pq and returned.

(a) Using Boundary Value Analysis and Functional Testing, identify two additional black-box test cases **for the remove() method**. The first test case is given to you as an example. (8 marks)

PQ (test case and test data)	Expected Output
Test case: PQ is empty Test data: {}	IllegalExceptionArgument thrown After remove(), PQ = {}, i.e. no change

Now consider the following Java implementation of PQ:

```
public class PQ
{
   private LinkedList list[]; // an array of LinkedLists
   private int highest;       // stores the highest priority value

   public PQ() {
      list = new LinkedList [10];
      highest = -1;
      for (int i=0; i<10; i++) { list[i] = new LinkedList(); }
   }

   public void append(Object o, int p) throws IllegalArgumentException{
      if (p < 0 || p > 9) throw new IllegalArgumentException();
      if (o == null) throw new IllegalArgumentException();
      if (p > highest) highest = p;
      list[p].addFirst(o);  // add to the front of list[p]
   }

   public Object remove() throws IllegalArgumentException {
      if (highest < 0) throw new IllegalArgumentException();
      Object o = list[highest].removeLast();// remove rear element
                                            // from list[highest]
      if (list[highest].isEmpty()){
           for (int i=highest-1; i>=0; i--)
                if (!list[i].isEmpty()) { highest = i; return o; }
           highest = -1;
      }
      return o;
   }
}
```

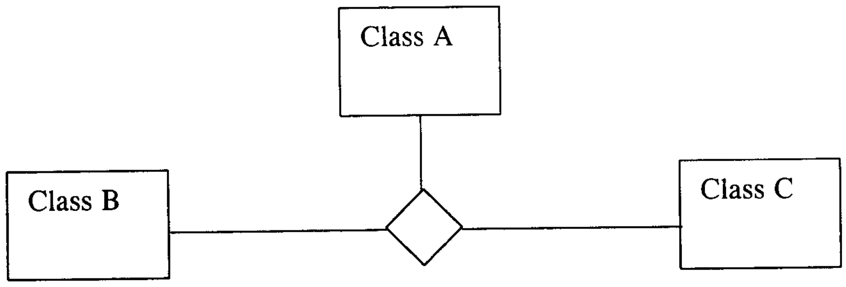
(b) Using white-box testing, identify three additional test cases **for the remove() method**. The first test case is given to you as an example. (12 marks)

PQ (test case and test data)	Expected Output

Test case: PQ is empty
Test data: {}
	IllegalExceptionArgument thrown
	After remove():
	list[i] = {}, for i= 0 to 9
	highest = -1

Question 5 (15 marks)

So far, we have been discussing class diagrams having binary relationships. Now suppose we need to model **tenary** relationships involving 3 classes, which is represented by the following notation:



(a) A large supermarket chain (e.g. NTUC, Cold Storage) owns multiple stores. It buys a wide variety of items from multiple suppliers. The Order Management System is responsible for placing orders to each supplier specifying the number of units of each item to be supplied to each of the stores. Using the above notation, draw a class diagram to model this aspect of the system. (5 marks)

(b) Can we replace a tenary relationship by 3 binary relationships? If so, show how that can be done with respect to the Order Management System. If not, explain why not. (5 marks)

(c) Refer to Figure 1. Is it appropriate to replace the 3 binary relationship with a tenary relationship? If so, show how this can be done. If not, explain your answer. (5 marks)

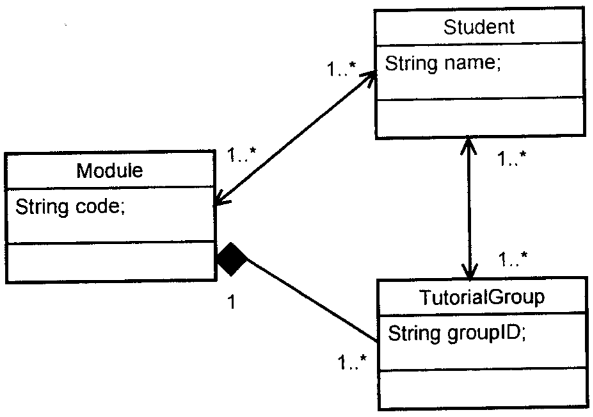


Figure 1. Module/Tutorial Management

END OF PAPER