NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
SEMESTER I (2008-2009)

MIDTERM TEST FOR
CS1102: DATA STRUCTURES AND ALGORITHMS

6.15pm, October 3, 2008
Time Allowed: 1 Hour

MATRICULATION NUMBER: HT040831R

INSTRUCTIONS TO CANDIDATES

1. Write your matriculation number in the space provided above. Shade your matriculation number on the OCR form. Remember to sign on the form
2. This examination paper consists of **2 sections**. Section 1 consists of **8 MCQ questions. Section 2 consists of 3 parts.**
3. **This examination paper** comprises **Eight (8)** printed pages including this front page.
4. Answer the MCQ questions using the OCR form and the other questions directly in the space given after each question. If necessary, use the back of the page.
5. Marks allocated to each question are indicated. Total marks for the paper is 100.
6. This is a closed book examination and you may write in pencil.

EXAMINER'S USE ONLY			
Section	**Possible**	**Marks**	**Check**
1	40		
2(a)	30		
2(b)	15		
2(c)	15		
Total	100		

Section 1. MCQ questions (5 marks each)

1) You are tasked to maintain a database of NUS students. There are two types of students, undergraduates and graduate students. About a third of the graduate students are doctoral candidates.

All of the students have the same personal information stored, like name, address, and phone number, and also student information like courses taken and grades. Each student's CAP is computed, but differently for undergraduates and graduates. The doctoral candidates have information about their dissertations and faculty advisors.

You have to write a java program to handle all the student information. Which of the following is the best design, in term of programmer efficiency and code reusability? Note: { … } denotes class code.

a) public interface Student { … } Unnecessary duplication here as interface cannot have method body
public class Undergraduate implements Student { … }
public class Graduate implements Student { … }
public class DocStudent extends Graduate { … }

b) public abstract class Student { … } There is no "student" class
public class Undergraduate extends Student { … }
public class Graduate extends Student { … }
public class DocStudent extends Graduate { … }

c) public class Student { … } → we do not want class student!
public class Undergraduate extends Student { … }
public class Graduate extends Student { … }
public class DocStudent extends Graduate { … }

d) public abstract class Student { … }
public class Undergraduate extends Student { … }
public class Graduate extends Student { … }
public class DocStudent extends Student { … }
↳ inappropriate

e) public interface PersonalInformation { … }
public class Student implements PersonalInformation { … } → we do not want class student
public class Undergraduate extends Student { … }
public class Graduate extends Student { … }
public class DocStudent extends Graduate { … }

2) Given the following player interface,

```
public interface Player {
     // Return integer that represents move in game
     int getMove();

     // Briefly describe strategy in choosing move
     void describeStrategy();
}
```

A class `HumanPlayer` implements the `Player` interface. Another class, `SmartPlayer`, is a subclass of `HumanPlayer`. Which statement is **FALSE**?

a) `SmartPlayer` automatically implements the `Player` interface.
b) `HumanPlayer` must contain implementations of both the `getMove` and `describeStrategy` methods.
c) It is not possible to declare a reference variable of type `Player`.
d) The `SmartPlayer` class can override the methods `getMove` and `describeStrategy` of the `HumanPlayer` class.
e) A method in a client program can have `Player` as a parameter type.

list has notion of "order" or "position"

3) In the ADT `list`, when an item is inserted into position `i` of the list, _____.

a) the position of all items is increased by 1
b) the position of each item that was at a position smaller than `i` is increased by 1
c) the position of each item that was at a position greater than `i` is increased by 1
d) the position of each item that was at a position smaller than `i` is decreased by 1 while the position of each item that was at a position greater than `i` is increased by 1
e) None of the above statements is true.

Single

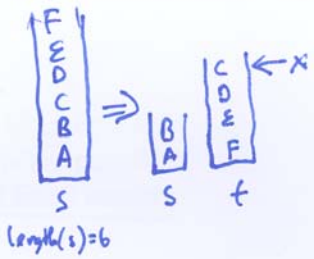
4) To delete a node N from a linear linked `list`, you will need to _____.

a) set the pointer `next` in the node that precedes N to point to the node that follows N ✓ (before)
b) set the pointer `next` in the node that precedes N to point to N ✗
c) set the pointer `next` in the node that follows N to point to the node that precedes N ✗
d) set the pointer `next` in N to point to the node that follows N ✗
e) None of the above statements is true.

5) Suppose that `s` and `t` are both stacks of `Object` and that `x` is a variable of type `Object`. Assume that `s` initially contains n elements, where n is large, and that `t` is initially empty. Assume further that `length(s)` gives the number of elements in `s`. Which is true after execution of the following code segment?

```
int len = length(s) - 2;
for (int j = 1; j <= len; j++) {
    x = s.pop();
    t.push(x);
}
len = length(s) - 2; -> len=0 here
for (int j = 1; j <= len; j++) {
    x = t.pop();
    s.push(x);
}
```

a) `s` is unchanged, and `x` equals the third item from the bottom of `s`.
b) `s` is unchanged, and `x` equals `s.peekTop()`.
c) `s` contains two elements, and `x` equals `s.peekTop()`.
d) `s` contains two elements, and `x` equals the bottom element of `s`.
~~e)~~ `s` contains two elements, and `x` equals `t.peekTop()`.



6) Suppose you are asked to implement a double-ended queue, which support addition and removal of items BOTH at the queue `front` and queue `tail`. Which of the following implementation is efficient (in terms of item shifting and/or nodes hopping)?

I. vector ✗ (not circular = inefficient)
II. Circular Singly Linked List with head pointer ✗ deletion from tail is difficult !
III. Circular Array ✓

a) II only
~~b)~~ III only
c) I and III only
d) I and II only
e) I, II and III

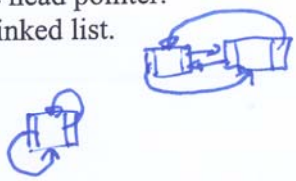
7) Given a circular doubly linked list with head pointer. A programmer wrote the following code for swapping the values of two adjacent nodes:

```
// Attempt to swap the values stored in
// node pointed by curPtr and its next node

temp = curPtr.item;
curPtr.item = curPtr.next.item;
curPtr.next.item = temp;
```

which of the following statement is **TRUE**? "fail" mens runtime error in the following statements.

a) The code will fail when curPtr points to the same node as the head pointer.
b) The code will fail when curPtr points to the last node in the linked list.
c) The code will fail when the linked list has two nodes only.
d) The code will fail when the linked list has one node only
e) None of the above is true



8) Given n integers stored in a normal queue (add at tail, remove at front). 0^{th} integer is at the front, $(n-1)^{th}$ integer is at the tail. Which of the following process is/are possible using ONLY the queue and no other data structure? Note that additional fixed number of integer variables is allowed.

I. Find the maximum integers
II. Move any k^{th} integer to the end of the n integers. End result is stored in the queue
III Reverse the n integers. End result is stored in the queue

a) I only
b) II only
c) I and II only
d) I and III only
e) I, II and III

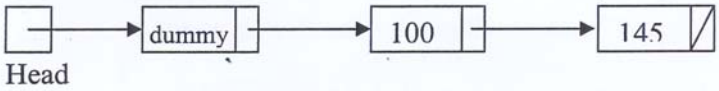
I. F B
[1, 5, 4] → 5, 4, 1 → 4, 1, 5 → 1, 5, 4
max = 1 max = 5 max = 5

II. 0 1 2
1, 5, 4 → 5, 4, 1 → 4, 1 → 1, 4 → 1, 4, 5
temp = 5

III. 1, 2, 3, 4 → 2, 3, 4, 1
2, 3, 4, 1 → 3, 4, 1 → 4, 1, 3 → 1, 3, 4 → 1, 3, 4, 2 → 3, 4, 2, 1
temp = 2
3, 4, 2, 1 → 4, 2, 1 → 2, 1, 4 → 2, 1, 4, 3 → 1, 4, 3, 2 → 4, 3, 2, 1
temp = 3

Section 2: Short questions (60 marks)

a) Both the insertion and deletion for linear linked list require a special case to handle action at the first position of a list. By adding a dummy head node as shown in the figure below, it eliminates the need for the special cases. Make the necessary changes to the following constructor and methods to take care of the dummy node. Write the codes in the boxes provided.




```
public class linkedList {

    public linkedList() {          // constructor: create empty linked list
        ListNode Head = null;
    }
```

new ListNode ("dummy", null);

10 marks

```
    public void insertAfter(ListNode current, Object item) {
        ListNode temp;

        if (current != null) {
            temp = new ListNode (item, current.next);
            current.next = temp; num_nodes++;
        }
        else
          // If current is null, insert item at beginning.
            head = new ListNode (item, head);
    }
```

10 marks

```
public void deleteAfter(ListNode current) throws ItemNotFoundException{

    if (current != null) {
        if (current.next!=null) {
            current.next = current.next.next; num_nodes--;
        }
     else
        throw new ItemNotFoundException("No Next Node to Delete");
    }
    else { // If current is null, assume we want to delete head.
        head = head.next; num_nodes--;
    }

}
```

10 marks

b) Suppose that you read a binary string – that is, a string of 0s and 1s – one character at a time. Describe how you could use a data structure you learned in CS1102 but no arithmetic (no counting) to see whether the number of 0s is equal to the number of 1s. (You only need to describe your algorithm and not implementing it in Java.)

This problem can be solved using stack

```
Stack s = new Stack();
while (true) {
  number = read next character;
  if (end of input)
    break;
  if (s.empty() || s.peek() == number)
    s.push(number);
  else
    s.pop();
}
```

Output true if s.empty() == true
false otherwise

15 marks

c) When these numbers are not equal, state how you could tell which character – 0 or 1 – occurs most frequently and by how much its count exceeds the other's. Again, you are not allowed to count during the process to determine which character occurs most frequently but you are allowed to count only at the end before outputting the answer. (You only need to describe your algorithm and not implementing it in Java.)

Do step b above

The character left in the stack and the size of the stack is the answer

15 marks

-- End of Paper --