

NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

EXAMINATION FOR
Semester II: 2001-2002

CS2103/TDCS2103 – SOFTWARE ENGINEERING

April/May 2002

Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **SIX (6)** questions and comprises **ELEVEN (11)** printed pages, including the cover page.
2. Answer **all questions**. Write your answers in the **blank spaces** in this answer book only.
3. Full mark of this paper is **100**.
4. This is an **OPEN BOOK** examination.
5. Please fill in your **Matriculation Number** below.

Matriculation Number: _____

For Examiner's Use Only	
	Marks
Question 1	
Question 2	
Question 3	
Question 4	
Question 5	
Question 6	
TOTAL:	

Question 1

8 marks

Consider an application at TrainWell Inc. , a small training company. For simplicity assume that company runs only one kind of workshop and has only one trainer. Each week of the training calendar is an instance of the class *WorkshopWeek*. The status of a given week is held in *WorkshopWeek.status* which can contain the values *available*, *reserved*, and *confirmed*. When a new *WorkshopWeek* is created its status is *available*. A customer can make a reservation for a given week , with or without a deposit. A deposit entitles the customer to confirm reservation; without it the customer gets only reservation. An inbound message *custReserve(amount)* causes a transition to either the reserve or confirmed status.

Draw a state diagram of three states of *status* described above. Label the transitions appropriately with firing events and guard conditions , if any.

Question 2

8 marks

Read the following scenario and draw a Class diagram.

When a new campaign starts, members of staff are assigned to work on it. A member of staff—a Director, an Account Manager or a member of the creative team—is nominated to be the contact for each client. A campaign can consist of many adverts. Details of each advert are entered into the system with a target completion date.

Include classes, associations and multiplicities in your diagram. You may ignore attributes and operations. Make reasonable assumptions to complete your diagram.

Question 3**14 marks**

Read the following requirements and draw a Use case diagram for a proposed on-line system for a PC retailer.

A PC retailer offers the possibility of purchasing PCs via the Internet. The customer can select a PC on the retailer's web page. The PCs are classified into HomeUse, OfficeUse and Gaming. The customer can select a standard specification (no optional extras) or can choose a desired specification on-line. The optional extras (such as GForce3 Graphics card) are presented as pick lists of available options. For each new specification, the on-line system can calculate price.

To place an order, the customer must fill out the shipment and payment information. Acceptable payment methods are credit cards and cheques. Once the order has been entered, the system sends a confirmation e-mail message to the customer with details of the order. While waiting for the arrival of the PC, the customer can check the order status on-line at any time.

The back end order processing consists of the steps needed to verify the customer's credentials and payment method, to request the ordered PC specification from the warehouse, to print an invoice and to request the warehouse to send the PC to the customer.

Extended Requirements :

1. The customer uses the retailer's on-line shopping web page to view the standard specification of the chosen PC. The price is also shown.
2. The customer chooses to view the details of the PC specification, perhaps with the intention of buying it as is or to request a more suitable specification. The price for each PC specification can be computed at the customer's request.
3. The customer may choose to order the PC on-line or may request that the salesperson contact him/her to explain order details, negotiate the price etc.
4. To place the order, the customer must fill out an on-line form with shipment and invoice address and with payment details (credit card or cheque).

{Use this space to answer Question 3}

Question 4**20 marks**

(a) Accountant and Waterskier are 2 classes. How would you make use of these 2 classes to define a class that describes accountants who like waterskiing? Justify your implementation.

(b) A, B, C, and D are classes such that A extends B, B extends C, and C extends D. When an object is instantiated from A, the default constructors of A, B, C and D will all be called. State the order in which these constructors are being called and briefly explain your answer.

4(c) Implement the exception `EmptyStackException`. Modify the `Stack` data abstraction given so that they will throw an `EmptyStackException` if an attempt is made to read or remove a value from an empty stack.

```
class Stack {
    private Vector theData;
    public Stack () {
        theData = new Vector();
    }
    public boolean empty() {
        return theData.isEmpty();
    }
    public Object push (Object item) {
        theData.addElement (item);
        return item;
    }
    public Object peek() {
        return theData.lastElement();
    }
    public Object pop () {
        Object result = theData.lastElement();
        theData.removeElementAt(theData.size()-1);
        return result;
    }
}
```

Question 5**20 marks**

A line (infinite in both directions), a ray (fixed at a point, infinite in one direction), and a segment (a portion of a line with fixed end points) are three geometric concepts.

(a) How would you structure classes representing these three concepts in an inheritance hierarchy?

(b) Characterize the form of inheritance you would use. Explain the reasoning behind your design.

5 (c) Would you change your design if you focus more on inheriting the data representation instead of the behavior, or vice versa? Show the changes needed as a result of the change in focus, or explain why there is no need to change the design of the classes at all.

Question 6**30 marks**

Draw the flow diagram for the following program. Identify a set of independent execution paths, test cases, and test data.

```

class CircularBuffer {
    int lastMsg,    // index of last message
    np,            // number of messages printed
    msgCounter;    // number of messages logged
    final int SIZE = 1000;    // Size of the message buffer
    String[] messageBuffer = new String[SIZE];

// ... OTHER METHODS
Line #
1      public int displayLastMsg(int nToPrint) {
2          np = 0;
3          if ( ( msgCounter > 0 ) && ( nToPrint > 0 ) ) {
4              for ( int j = lastMsg; ( j != 0 ) && ( np < nToPrint ) ; --j) {
5                  System.out.println(messageBuffer[j]);
6                  ++np;
7              } // end for
8              if (np < nToPrint) {
9                  for (int j = SIZE; ( j != 0 ) && ( np < nToPrint ) ; --j) {
10                     System.out.println(messageBuffer[j]);
11                     ++np;
12                 } // end for
13             } // end if
14         } // end if
15         return np;
16     } // end displayLastMsg

```

{Use this space to answer Question 6}

--END OF PAPER--