NATIONAL UNIVERSITY OF SINGAPORE
SCHOOL OF COMPUTING

SEMESTER I (2003-04)
EXAMINATION FOR

CS2103: SOFTWARE ENGINEERING

November 2003 Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper consists of **NINE (9)** questions and comprises of **THIRTEEN (13)** printed pages.
2. Write your answers in the **blank spaces** in this answer book only.
3. This is an **OPEN BOOK** examination.
4. Please fill in your **Matriculation Number** below. **Also write your matriculation number on the top right hand corner of every page.**

Matriculation Number: □□□□□□□□□□

For Official Use Only:

	Marks
Question 1 (max 10)	
Question 2 (max 5)	
Question 3 (max 15)	
Question 4 (max 10)	
Question 5 (max 10)	
Question 6 (max 5)	
Question 7 (max 5)	
Question 8 (max 15)	
Question 9 (max 25)	
TOTAL:	

Question 1 **10 points**

(a) What do you think is the link between O-O approach of development and prototyping?

(b) From an O-O point of view, the Waterfall model has a major drawback. What do you think it is?

Questions 2- 5 are based on the following requirements.

A library wishes to develop a new library system. The requirements of this system are as follows:

- Library loans books and magazines to borrowers, who are members of the library.
- The librarian is an employee of the library who interacts with the members (borrowers) and whose work is supported by the system
- Library purchases new titles for the library. Popular titles are bought in multiple copies. Old books and magazines are removed when they are out of date or in poor condition.
- A borrower can reserve a book or magazine that is not currently available in the library, so that when it's returned or purchased by the library, that member is notified. The reservation is cancelled when the borrower checks out the book or magazine or through an explicit canceling procedure.
- Library can easily create, update, and delete information about the titles, borrowers, loans, and reservations in the system.
- Borrowers can search for titles and make/cancel reservations online via a website.

When items are being loaned or returned the borrower interacts with the librarian. The librarian can also reserve titles and cancel reservations *on* behalf of the borrower. Librarians can search for items/titles in the library. If a borrower has a reservation on an item then this reservation is *removed* when the item is loaned to him.

Write your Assumptions / Interpretations, if any, here:

Question 2 5 points

Produce a *use case diagram* for part of the new system being developed. Model the part of the system that deals with loaning ,returning , reservations, and searching of items.

Question 3 **15 points**

Produce a *class diagram* for part of the system being developed. Include the classes and information given below. Find relationships (association, specialization/ generalization, aggregation) between these classes. Add multiplicities of associations, and attributes (along with appropriate types) needed in each class. Feel free to add any additional classes where required.

The classes to be included are: Title, Magazine, Book, Item, Loan, Borrower, Reservation

These classes must include the following information:

1. All Titles have a unique ISBN and store an array of the Items of this Title in the library (Items represent physical copies of a Title) and an array of the Reservations that exist for this Title.
2. Magazine has a magazine name, editor, and issue number.
3. Book has a book name, and author.
4. Item has a barcode and store the Title that it is a copy of as well as the Loan which contains information about who it is currently loaned to.
5. Loans consist of Borrower and Item.
6. Borrower stores the information about a single borrower. It stores first and last names, address, city, and postcode. It also stores an array of current Loans and an array of current Reservations.
7. Reservation consists of Title and Borrower.

{Additional Space for Question 3 }

Question 4 10 points

Produce an interaction diagram to model the use case 'Loan Item'. This should show the sequence of messages between objects of the following classes and their methods.
Feel free to add any additional classes or methods where required.

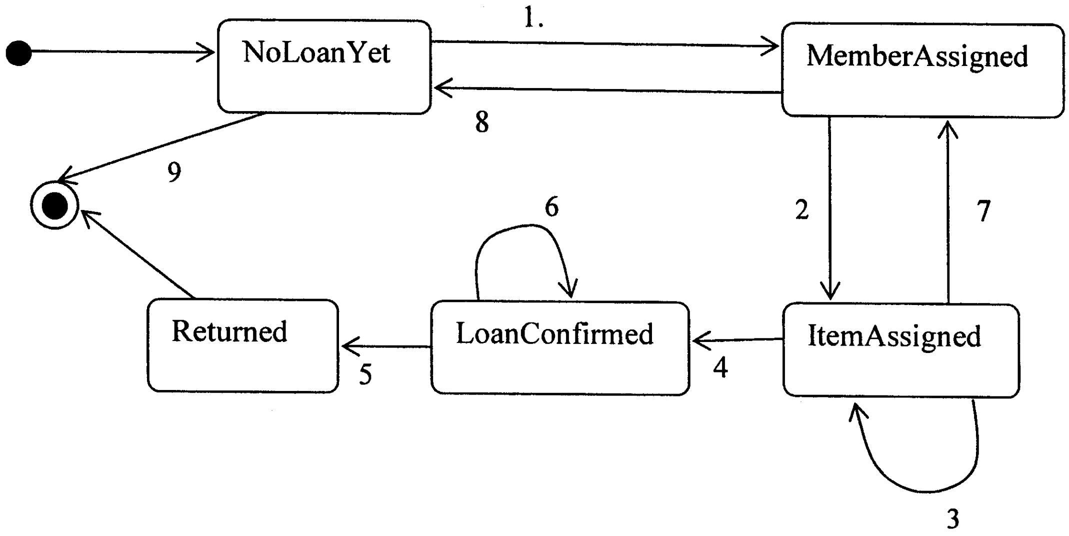
class	**method**
Loan	newLoan (Item, Borrower)
Reservation	removeReservation (Title)
Item	getTitle( )
Title	removeReservation (Borrower)
Borrower	removeReservation (Title)

Question 5 **10 points**

Given below is a *StateChart Diagram* that models the lifecycle of a **Loan** object. On creation, Loan object is in 'NoLoanYet' state. Once the member and item is assigned to it, Loan object goes into 'LoanConfirmed' state. Transitions on the diagram are labeled with digits 1 to 9. Complete the table that follows the diagram with appropriate events which should label the transitions of the diagram along with any guard conditions you find appropriate. A few of the cells in the table are filled for you.

Possible Events: newLoan, cancel , removeReservation, renewLoan

Feel free to add any additional events.



Label Number	Event	Guard, if any
1	newLoan	Nil
2	newLoan	Borrower Loans < MAX
3		
4	newLoan	Nil
5		Nil
6		
7		
8		
9	Cancel	

Question 6 **5 points**

What will the following program print given the fact that a `clone` of a multidimensional array is shallow? Explain the result.

```
class Test {
  public static void main(String[] args) throws Throwable {
       int ia[][] = { { 1 , 2}, null };
       int ja[][] = (int[][])ia.clone();
       System.out.print((ia == ja) + " ");
       System.out.println(ia[0] == ja[0] && ia[1] == ja[1]);
       }
}
```

Question 7 **5 points**

```
interface Fish { int getNumberOfScales(); }
interface StringBass { double getNumberOfScales(); }
class Bass implements Fish, StringBass {
      public ??? getNumberOfScales() { ??? }
}
```

Complete the implementation of Bass by replacing ??? with appropriate expressions. Briefly explain what you have implemented.

Question 8 **15 points**

Given the finite state machine (diagram) of a turnstile as follow:

State	event/action	end state
Locked	pass/alarm	Locked
Locked	coin/unlock	Unlocked
Unlocked	coin /thankyou	Unlocked
Unlocked	pass/lock	Locked

Use State pattern to implement the class Turnstile and whatever relevant classes and interface needed when State pattern is used.

Question 9 **25 points**

Implement the class Stack as a subclass of Vector to manipulate objects. A typical stack contains the following operations:

void push(Object obj) – pushes **obj** onto the top of the stack

Object pop() – removes and return an object from the top of the stack.
Throws StackEmptyException if the stack is empty.

Object peek() –return an object without removing it from the top of the stack.
Throws StackEmptyException if the stack is empty.

boolean empty() – returns true if the stack is empty

(a) **[10 points]** Write the code for the class Stack, including the class StackEmptyException.

(b) **[5 points]** Write the test script for testing the classes implemented

(c) **[10 points]** Write the test driver to test the classes. It should make use of TDIO, the class that handles the IO of a test driver, to read from the test script you have just prepared, test whatever methods in Stack, handle exceptions properly, check the results and collect the test statistics.

{Additional space for Question 9 }

- End of Paper -