NATIONAL UNIVERSITY OF SINGAPORE
SCHOOL OF COMPUTING

SEMESTER II (2006-07)
EXAMINATION FOR

CS2103: SOFTWARE ENGINEERING

April 2007 Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper consists of **SEVEN (7)** questions and comprises **fourteen (14)** printed pages. Answer **ALL** questions.
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 (i) to (x) (max 20)	
Question 2 (max 10)	
Question 3 (max 15)	
Question 4 (max 15)	
Question 5 (max 10)	
Question 6 (max 10)	
Question 7 (max 20)	
TOTAL: (max 100)	

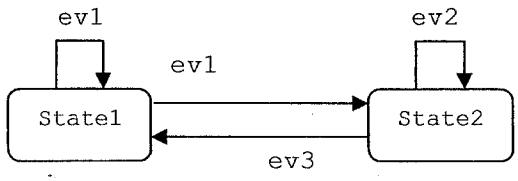
Question 1: It has 10 multiple choice questions (MCQ). Each MCQ is evaluated with 2 marks. Fill answers for these MCQs in the following table.

Question	(i)	(ii)	(iii)	(iv)	(v)	(vi)	(vii)	(viii)	(ix)	(x)
Answer										

Question (i). What is the difference between the activity diagrams and statecharts?

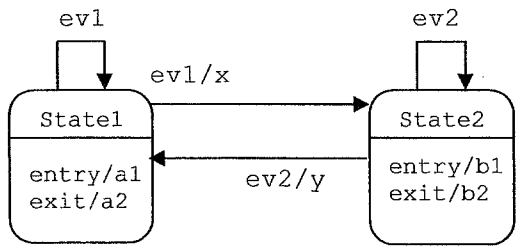
a) Both activity diagrams and statecharts specify the behaviour of more objects in relation to events;
b) Activity diagrams are the same thing as interaction diagrams, and statecharts do not specify all the possible behaviours of objects;
c) Activity diagrams do not normally include events, whereas statecharts summarize the overall behaviour of objects in relation to events;
d) Both activity diagrams and statecharts specify activities that can get stuck;

Question (ii). What is NOT true about the below statechart?



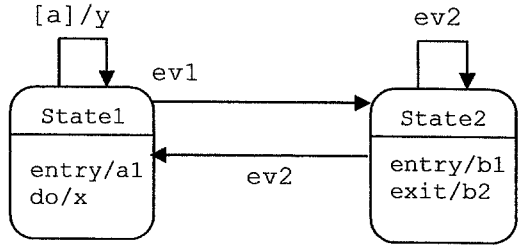
a) The transitions labeled with `ev1` are non-deterministic;
b) The transition labeled with `ev2` is a self-transition;
c) `State1` and `State2` can be active at the same time;
d) As a response to event `ev1`, the statechart can either stay in `State1` or go to `State2`;
e) None of the above.

Question (iii). What is true about the below statechart?



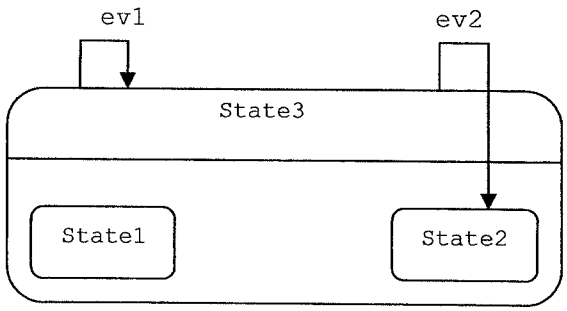
a) The guards `/x` and `/y` are missing from the self-transitions;
b) The activity `x` has the role to go from `State1` and `State2`;
c) The entry action `b1` will be executed when entering `State2`;
d) The exit activity `a2` will be executed when entering `State2`;
e) None of the above.

Question (iv). What is true about the below statechart?



a) The action `x` is performed over an extended period of time;
b) The activity `x` is performed instantaneously and cannot be interrupted by `ev1`;
c) The completion transition `y` can be interrupted by event `ev1`;
d) The completion transition `y` can be interrupted by event `ev2`;
e) None of the above.

Question (v). What is true about the below statechart?



a) The transition labeled with `ev1` is incorrect;
b) The transition labeled with `ev2` is incorrect;
c) There is no self-transition of `State1`;
d) There is a transition from `State1` to `State2`;
e) None of the above.

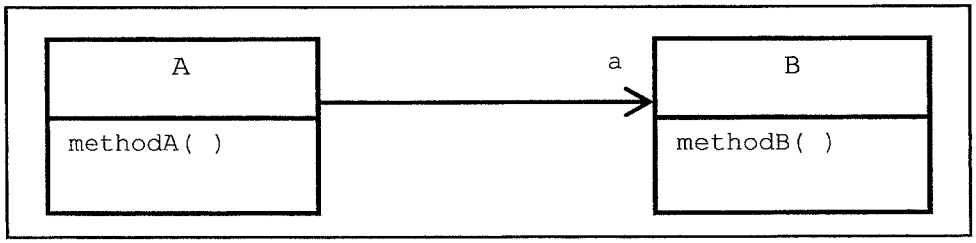
Question (vi). What is the proper design pattern to be applied when implementing statecharts?

a) The Abstraction-Occurrence Pattern, as statecharts refer to a set of related objects that share common information;
b) The General-Hierarchy Pattern, as statecharts refer to a set of objects that have a naturally hierarchical relationship;
c) The Player-Role Pattern, as statecharts refer to a set of objects that may play different roles in different contexts;
d) The State Pattern, as statecharts refer to a set of objects supposed to alter their behaviour when their internal states change;
e) None of the above.

Question (vii). What is true regarding the `Observer` pattern?

a) `<<Observer>>` must implement the method `update();`
b) Changes to the classes that implement `<<Observer>>` do not affect the observable classes;
c) There is a many-to-many dependency between the observable classes and the `<<Observer>>` abstract class;
d) Whenever there is a change in the observers, all observables will be notified;
e) None of the above.

Question (viii). What is true when doing the implementation of the following class diagram?



a) The implementation of `A` should contain an attribute `a` of class `B`;
b) The implementation of `B` should contain an attribute `a` of class `A`;
c) The implementation of `methodB()` should contain the reference `a`;
d) Class `B` makes use of class `A`;
e) None of the above.

Question (ix). What is true regarding the top down and bottom up implementations?

a) Top down strategies must postpone the generation of complete executable program;
b) High level classes can usually stand alone;
c) Bottom up strategies need temporary stubs for lower classes;
d) Starting with high level component enables early system testing;
e) None of the above.

Question (x). What is true regarding the black box and white box testing?

a) The black box and the white box testing are actually the same thing;
b) For black box testing, tests are conducted at the interface of the component;
c) For black box testing, tests know the internal workings of a component;
d) For white box testing, tests do not know the internal workings of a component, but know only the specified function a component has been designed for;
e) None of the above.

Question 2 (10 marks) From the following description, describe the use case, by identifying the actors, basic course of events and possible alternatives.

"In order to exit from the bus, each passenger should pay by EZ-Link or by cash. Once the payment has been successfully done, the passenger can exit the bus."

(Space for the answer of question 2)

(Space for the answer of question 2)

Question 3 (15 marks) From the following description, identify the main objects and their links. Draw a class diagram that specify the below description. Be sure to indicate the multiplicity, roles and name of each association. Also, draw an object diagram that shows these objects and links. Justify your choices.

"SingTel, M1, Maxis, and Celcos are all mobile phone companies. SingTel and M1 have coverage in Singapore whereas Maxis and Celcos have coverage in Malaysia. There are many customers of the above mobile phone companies. A customer can be a client of more than just one mobile phone company."

(Space for the answer of question 3)

(Space for the answer of question 3)

Question 4 (15 marks) The common information of a set of objects is placed in the class `Base` and the specific information of the objects in class `Derived`. Let us consider the following Java program:

```
class Base {
      private int a;
      public Base(int a) { this.a = a; }
      public void set(int a) { this.a = a; }
      public int getA() { return a; }
}

class Derived extends Base {
      private int b;
      public Derived(int a, int b) {
            super(a);
            this.b = b;
      }
      public int getA() { return super.getA(); }
}
```

(a) (5 marks) Draw a class diagram valid with the above Java program;

Let us now consider the following main Java program that uses the above classes:

```
public class Exam {
      public static void main(String args[]) {
            Base objBase = new Base(1);
            System.out.println(objBase.getA());
            Derived objDerived = new Derived(1,2);
            System.out.println(objDerived.getA());
            objBase.set(3);
            System.out.println(objBase.getA());
            System.out.println(objDerived.getA());
      }
}
```

(b) (5 marks) Draw a sequence diagram that shows the execution of the above statements.

(c) (5 marks) Since the changes of the attributes of class `Base` are not visible for class `Derived` in the above Java program, redesign the previous class diagram such that the common information of the objects will be represented without duplications by avoiding inconsistency when changing the common information. Which design pattern do you think it is the best for that?

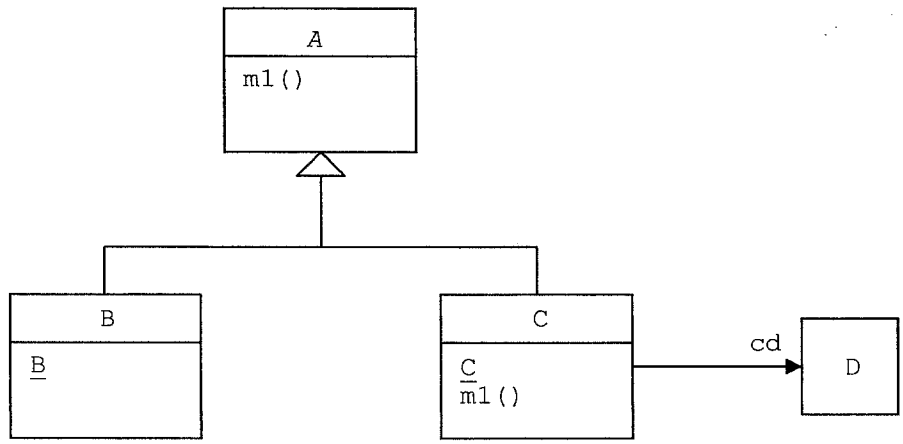
(Space for the answer of question 4)

(Space for the answer of question 4)

Question 5 (10 marks) Let us suppose you have an application that needs just one object (instance) created for a given class (e.g., there is only one university called 'National University of Singapore'). Which is the design pattern proper to be used in this case? Write a Java code so that your final program will respect this requirement. Show also what happens when you try to create two objects of that class. Illustrate a way to access the private attributes of that class.

(Space for the answer of question 5)

Question 6 (10 marks) Write a correct Java program that implements the below class diagram. Try to write as much complete code as possible.



(Space for the answer of question 6)

Question 7 (20 marks) Read the following Java code fragment for a method which attempts to calculate the number of students having the mark between 1 and 4.

(i) Draw a control flow graph.
(ii) Based on the control flow graph, find the cyclomatic complexity and list all the (linearly) independent paths.
(iii) Prepare a test case for each independent path.

```
ns = 0;
I = 1;
while (I <= N) {
  I++;
  if (mark[I] >= 1.0 && mark[I] < 4.0)
    ns++;
}
print ns;
```

(Space for the answer of question 7)

(Space for the answer of question 7)

---END OF PAPER--