NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING
EXAMINATION FOR
Semester 1: 2002/2003

CS2103 – SOFTWARE ENGINEERING

November 2002 Time Allowed 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **SEVEN (7)** questions and comprises **TWELVE (12)** 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. **Please write your Matriculation Number Below.**

MATRICULATION NO: ________________________________

This portion is for examiner's use only

QUESTION	MARKS	REMARK
1 (10 marks)		
2 (15 marks)		
3 (15 marks)		
4 (25 marks)		
5 (10 marks)		
6 (15 marks)		
7 (10 marks)		
Total		

Question 1 **10 marks**

Consider the requirements of EMS system developed during the course CS2103. A copy of the requirements are also attached here as **Appendix A** on the last page of this examination booklet. Consider two of the classes ***Task*** and ***Resource***, identified during analysis and design activity of the EMS system. Objects of the type of ***Task*** and ***Resource*** undergo changes in their respective state due to certain events. Analyze the requirements to identify the states, and the events that trigger the changes in the state of these two objects. Present your analysis using **state-chart diagrams** for objects of the type ***Task*** and ***Resource,*** clearly indicating the states, and events that trigger transition from one state to another state for each of these objects. You may choose any reasonable and context relevant labels for labeling states and transitions.

Question 2 **15 marks**

Read the following description and draw a **class diagram** identifying any specialization-generalization, whole-part, or simple associations between these classes. Label all associations, and indicate multiplicities. You are not required to add any attributes or operations.

A component is a part of the system that can be isolated for testing. A component can be an object, a group of objects, or one or more subsystems. A fault, also called bug or defect, is a design or coding mistake that may cause abnormal component behavior. An error is a manifestation of a fault during the execution of the system. A failure is a deviation between the specification of a component and its behavior. A failure is triggered by one or more errors. A test case is a set of inputs and expected results that exercises a component with the purpose of causing failures and detecting faults. A test stub, a part of test case, is a partial implementation of a component that depends on the tested component. A correction is a change to a component. The purpose of a correction is to repair a fault.

Question 3 **15 marks**

An n-step counter is a counter whose value is incremented and decremented by a fixed integer, called its step. An NStepCounter object is initialized with its step when it is constructed. Its ***inc*** method increments the counter by its step, its ***dec*** method decrements the counter by its step, its ***step*** method reports its step, and its ***value*** method reports its current value.

(a) Implement the class NStepCounter.

(b) Now use inheritance for extension to define a new class ClearableCounter that offers the same services as NStepCounter, but which also provides a ***clear*** method that resets the counter's value to zero. Implement this class ClearableCounter.

Question 4 **25 marks**

(a) Complete the implementation of module registration example (from lecture notes) with the use of an association class.

(b) Write the code to find the average mark of a given module.

(c) Write the code to find the average mark of all modules read by a student.

(d) Draw a collaboration OR sequence diagram for the part (c) above, clearly showing the objects and object interactions.

Question 5 **10 marks**

Use a design pattern to describe the relationship between classes required to implement a UNIX directory structure. In this structure, there are directories, sub-directories, and files. Write down the name of the pattern, and draw a class diagram of the directory structure with the use of the pattern, and a brief explanation of your solution.

Question 6 **15 marks**

Consider the following method:

```
static int maxOfThree (int x, int y, int z) {

   if (x > y)
      if (x > z) return x; else return z;
   if (y > z) return y; else return z;

}
```

(a) Draw a flow graph corresponding to the method given above.
(b) List all the execution paths.
(c) Construct a **smallest** set of test data to cover all the execution paths.

Question 7 **10 marks**

(a) What is the return type of a constructor?

(b) Consider ***a*** is an array and ***s*** is a string. State **True** if the statement given below is legal, otherwise state **False** in the parentheses next to the statement.

(i) Object o1 = *a*; [ ]

(ii) Object o2 = *s*; [ ]

(c) Choose the most correct statement from the following.

In the java statement "Ball football = new Ball()",

(i) the Ball() method should be called recursively as Ball appears twice
(ii) a Ball object is created and stored in football
(iii) a Ball object is created and its reference is stored in football
(iv) a Ball object is created from the part of memory that has not been used so far

Answer : ( )

(d) Give an example of a program fragment to show that the use of a certain test data that ensure all statements being executed at least once fails to ensure the full coverage of all testing conditions.

(e) Explain why "Child override" is printed in the following program.

```
class Parent {
  void overload (Parent p) {
    System.out.println("Parent");
  }
}

class Child extends Parent {
  void overload (Parent p) {
    System.out.println("Child override");
  }
  void overload (Child c) {
    System.out.println("Child overload");
  }
  void testOverload ( ){
    Parent p = new Child();
    overload(p);
  }

  public static void main(String[] args) {
    Child c = new Child();
    c.testOverload();
  }
}
```

Appendix A : EMS system requirements for Question 1

Consider a computer aided Emergency Management System (EMS). Field officers such as a policeman access to a wireless computer that enables them to interact with the EMS. On receiving an emergency call, the Call Taker (or Field Officer) creates a task and captures its type, time, urgency, resource required, and description. Once a task is created the, task map is updated to reflect the newly created task in accordance to the task location. As an example, Task map could be represented on a geographical map with symbols to show active tasks and a description could be popped up on a mouse-click on a particular task. The system alerts Dispatcher on any new task creation. A Dispatcher can view a list of un-dispatched tasks, anytime. A Dispatcher can also visualize the current states of all its Resources, such as, police cars, on a screen and dispatch a resource by issuing commands from his workstation. The Dispatcher selects a number of free Resources, and dispatches them to execute the selected task. The task's state is changed to reflect 'attended, and the task map is updated to reflect the new status (i.e. symbol/color change) . A task is set to 'closed' status on its completion. All Resources that are associated with the task are cleared, and the task map is updated.

-----END OF PAPER------