NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

EXAMINATION FOR SEMESTER 2 AY2005/2006

CS2103 – SOFTWARE ENGINEERING

April 2006 Time Allowed: 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **SIX (6)** questions and comprises **FIFTEEN (15)** printed pages, including this page.
2. Answer **ALL** questions within the space in this booklet
3. This is an **OPEN BOOK** examination.
4. Please write your Matriculation Number below.

MATRICULATION NO: ______________________________

This portion is for examiner's use only

Question	Marks	Remarks
Q1 (max = 12)		
Q2 (max = 10)		
Q3 (max = 9)		
Q4 (max = 6)		
Q5 (max = 9)		
Q6 (max = 4)		
Total (max = 50)		

Question 1 (12 marks)

Suppose you are building a software system for a **clinic**. The basic functionality required is to keep track of the **patients** and their **consultation records**. A new consultation record is created whenever a patient visits the clinic. This new record will be kept along with other older consultation records for that patient.

A **patient record** contains:

- Name of the patient
- NRIC of the patient
- A number of **consultation records**

A **consultation record** contains:

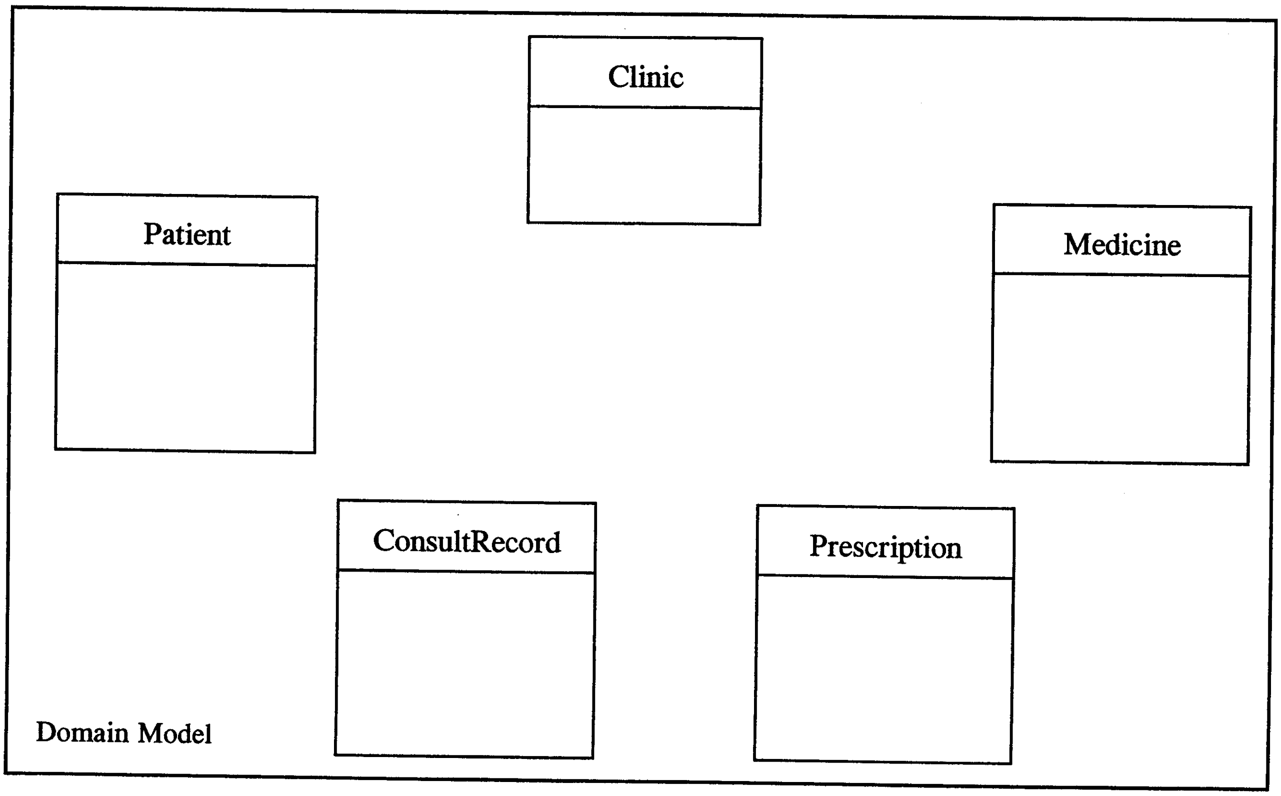
- Date of the consultation
- Description of the sickness (can assume to be some text)
- Not more than three medicine **prescriptions**

A **prescription** contains:

- Name of the medicine
- Manufacturer of the medicine
- Quantity (assume all medicines are pills)

Below is a partial domain model for the problem domain. The abstraction-occurrence design pattern has been applied to reduce redundant information, which results in the **medicine-prescription** classes.

Part I) Add in relevant attributes and associations to complete the diagram. You should indicate multiplicity and association name clearly. **(2 marks)**



(**Question 1** continues on next page)

Question 1

Part II) Using the domain model in Part I, draw an **object diagram** for the following scenario. (**3 marks)**

A patient "Nefos" with NRIC no "S1234Z" has visited the clinic twice. The first consultation, dated "1-January-2006" and he is diagnosed with "Fever and Headache". The prescription given was "ACP 10 pills, IPD 20 pills". The second consultation was on "15-Febuary-2006", where he is diagnosed with "Flu". The prescription given was "DSN 12 pills, ACP 20 pills and THX 10 pills". The manufacturer of the medicines is listed below for your reference:

- ACP by VembraCorp
- DSN by Berjame Medicine
- IPD by Sunimoto Chemical
- THX by VembraCorp

You can use the following abbreviations in the diagram to save some time:

- CRec = ConsultRecord
- Pres = Prescription
- Med = Medicine

(**Question 1** continues on next page)

Question 1

Part III) Given the following use case description, draw a **sequence diagram** to realize the use case. Boundary classes (UI classes) can be ignored. The diagram should base on the classes in the domain model in Part I. You are allowed to introduce new class(es) where necessary. (**3 marks**)

Display Consultation History: Basic Course Of Events:

1. User enters NRIC of a patient
2. System displays **all** consultation records for the patient, which includes:
   a. Date of consultation
   b. Medicine(s) Name and Quantity

Note that abbreviations introduced in Part II can be used to save some time.

(**Question 1** continues on next page)

Question 1

Part IV) Given the following use case description, draw a **collaboration diagram** to realize the use case. Boundary classes (UI classes) can be ignored. The diagram should base on the classes in the domain model in Part I. You are allowed to introduce new class(es) where necessary. (**4 marks**)

Add New Consultation Record: Basic Course Of Events:

1. User enters the following information:
    a. NRIC of patient (assume the patient exist in the system)
    b. Current Date
    c. Description of sickness
    d. Medicine prescribed (assume only medicine already in system will be used)
        i. Name and Quantity of medicine
2. System records the information.

State the property and the association role stereotype clearly for all association roles. Label your messages with hierarchical sequence number. Abbreviation in Part II can be used to save some time.

Question 2 (10 marks)

Part I) Give **TWO** conceptual differences (**not** notational differences) between the *General-Hierarchy Pattern* and the *Composite Pattern*. Be concise in your answer. (**4 marks**)

Part II) In both *General-Hierarchy Pattern* and the *Composite Pattern,* there is a subclass with an association to the superclass. In *Composite Pattern,* that particular association is **always** represented as an aggregation. However, in the *General-Hierarchy Pattern*, that relationship is usually represented as a normal association. Give a reason for this from the **implementation** point of view. (**2 marks**)

(**Question 2** continues on next page)

Question 2

Part III) Draw a class diagram for the following description. Apply design pattern(s) where appropriate. (**4 marks)**

Items in a convenient store contain the following information:

- Name
- Manufacturer
- Price
- Bar Code (unique to each item)

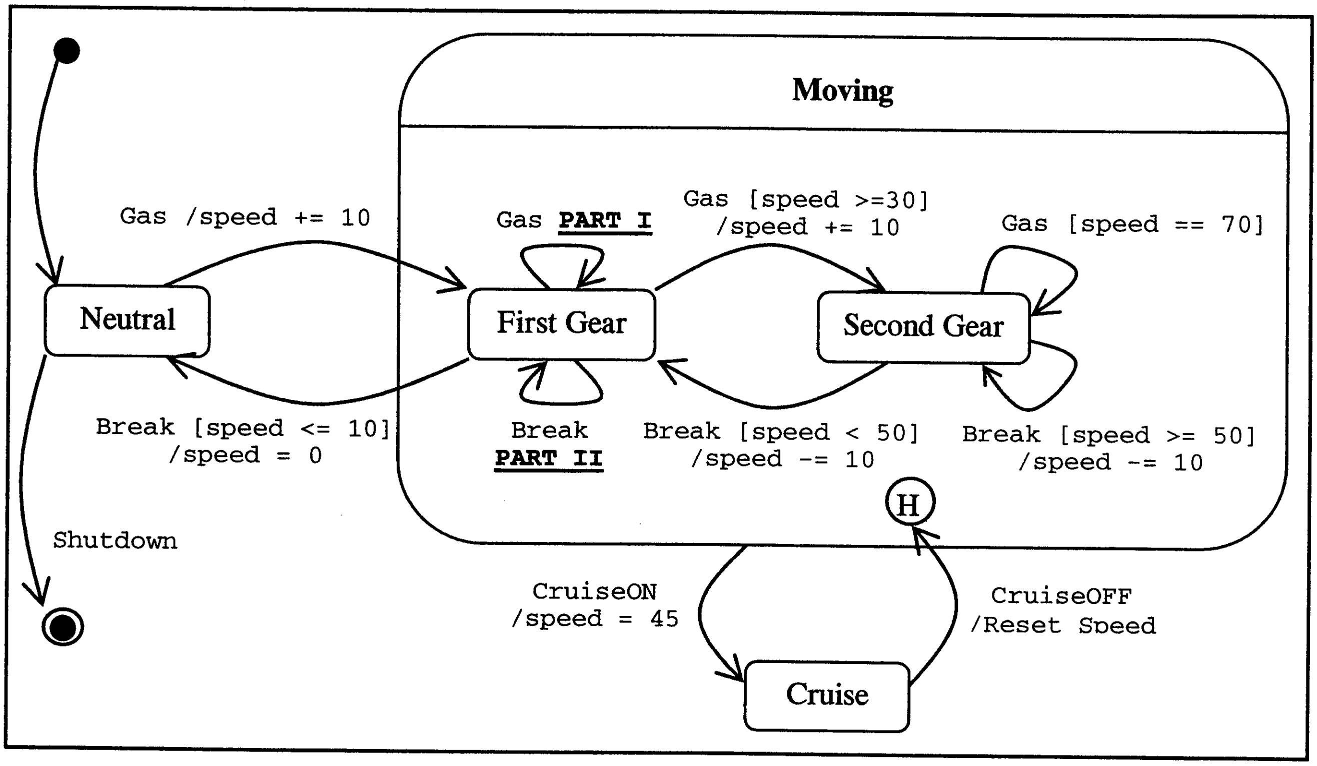
Also, an item can be in the following states:

- Ordered
- In storage
- On Display

Relevant attributes and operations for the classes should be included.

Question 3 (9 marks)

The startchart below describes the states of a car engine with automatic transmission (i.e. changes gear automatically). The engine starts off in the "Neutral" state with **speed = 0**. Whenever the driver hit the gas pedal (represented as **"Gas"** trigger in the statechart), the speed will increase by 10 until it hit the maximum of 70. The break pedal (represented as **"Break"**) slows the car down by 10. Whenever the car speed hit 40, the engine goes into "Second Gear" state. Likewise, whenever the speed drop below 40, the engine retreats to "First Gear" state. During these two states, the driver can activate the **Cruise Control** (represented as "CruiseON") which fix the speed at 45. In the "Cruise" state, gas and break pedals cease to function. When the driver deactivates the **Cruise Control,** the engine **will reset to the speed before the cruise control activation**. For example, if the driver activates the cruise control when car moving at 60. Then the engine will restore the speed to 60 when the cruise control is turned off.



Part I) Complete the transition "Gas **Part I**" for the "First Gear" state: **(1 mark)**

Gas ______________________________

Part II) Complete the transition "Break **Part II**" for the "First Gear" state: **(1 mark)**

Break ______________________________

(**Question 3** continues on next page)

Question 3

Part III) Given the class skeleton below, implement the following: (**7 marks**)

- Declaration of relevant attributes
- Constructor
- The *Gas( )* method for handling the "Gas" transitions
- The *CruiseOn( )* method for the "CruiseON" transition
- The *CruiseOff( )* method for the "CruiseOFF" transition

```
class Engine
{

//Declaration of relevant attributes










//Implementation of Constructor
Engine(   )
{










}
//............... Continue on next page
```

```
//Implementation of the "CruiseON" transition

public void CruiseOn( )
{









}

//Implementation of the "CruiseOFF" transition

public void CruiseOff( )
{









}
//............... Continue on next page
```

```
//Implementation of the "Gas" transitions
public void Gas( )
{

}


} // End of Class Engine
```

Question 4 (6 marks)

Given the following Java code fragment :

```
class A
{

public void M(B ob){
      System.out.print("MA:  ");
      ob.printInfo( );
}

public void printInfo( ){
     System.out.println("A");
}

}
```

```
class C extends B
{
public void M(C oc){
      System.out.print("MC:  ");
      oc.printInfo( );
}

public void printInfo( ){
     System.out.println("C");
}
}
```

```
class B extends A
{

public void printInfo( ){
     System.out.println("B");
}

}
```

Assume that we have the following declarations:

```
A refA, objA = new A( );
B refB, objB = new B( );
C refC, objC = new C( );
```

Fill in the following table with relevant information:

Java statements	Error (State "compile time", "runtime error" or "no error")	Output if applicable
`refA = ( B )objC;`		
`refC = ( B )( A )objC;`		
`( (C)objB ).M( objC );`		
`( (B)objC ).M( objC );`		
`objC.M( (B)objC );`		
`( (B)objC ).M( (B)objC);`		

Question 5 (9 marks)

Given the pseudo code fragment below for calculating leap year:

```
boolean isLeapYear(int year)
{
       boolean leapYear = false;

       if (year < 0 )
            return false;

       if (year % 4 == 0)
            leapYear = true;

       if ( ( year % 100 == 0) && (year % 400 != 0) )
           leapYear = false;

       return leapYear;

}
```

Part I) Draw control flow graph for the code fragment. **(3 marks)**

Part II) Give the cyclomatic complexity number for the control flow graph in Part I). **(1 mark)**

(**Question 5** continues on next page)

Part III) List the independent basic paths, test data and expected output (the return value of the method) for each path. Note that you should provide as many paths as dictated by the cyclomatic complexity number only. Do **NOT** take the space provided as indication of possible answers. (**5 marks**)

	Basic Paths	**Test Data**	**Expected Output**
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			

Question 6 (4 marks)

In University of Nowere, students are given matriculation number according to the following format:

```
[Faculty Alphabet] [Gender Alphabet] [Serial Number] [Check Alphabet]
```

The valid value(s) for each part of the matriculation number is given below:

Faculty Alphabet:

- Single **Capital Alphabet**
- Only **'C'** to **'G'** are valid

Gender Alphabet:

- Single **Capital Alphabet**
- Either **'F'** or **'M'** only

Serial Number

- 4 digits number
- From **1000** to **9999** only

Check Alphabet

- Single **Capital Alphabet**
- Only **'K'**, **'H'**, **'S'**, **'X'** and **'Z'** are valid

Fill in the following table:

	Equivalence Partitions	**Sample Test Data (Give Boundary Values if applicable, otherwise give one data per partition)**
Faculty Alphabet		
Gender Alphabet		
Serial Number		
Check Alphabet		

**~~~~END OF PAPER~~~~**