CS 3415 FINAL EXAMINATION

December 8th, 2008

Duration: 3 hours

Student Name:

Student Number:

Note: There are total 15 pages in this exam set. Please check the page numbers before you start to solve questions.

Problem 1 (10 marks)

Fill `Y/N` in each of the following brackets to indicate the related statement is correct or incorrect.

1. Having performed domain analysis will faster the development of a system. ( )
2. One of the most important things about quality requirements is to make them verifiable. ( )
3. A scenario is a special type of use case. ( )
4. Directionality of an association between classes should be defined in the beginning when you consider an association. ( )
5. An object diagram shows the relationships among the objects that will exist at run-time. ( )
6. In UML, a user interface is usually specified as a small circle (like a lollipop). ( )
7. An attribute in a class can normally represent a variable number of things. ( )
8. The Abstraction-Occurrence pattern is most frequently used in class diagram that form part of a system domain model. ( )
9. One principle of usability of user interface is put as much as information in the GUI to attract users. ( )
10. A communication diagram shows several objects working together with communication links instead of links of associations. ( )

Problem 2 (20 marks)

Answer following questions:

1. What is a scenario in developing requirements?

2. What is an association between two classes?

3. What is the Façade pattern?

4. What is the Singleton pattern?

Problem 3 (20 marks)

A use case for a university registration system is briefly described as follows. The name of the use case is **Register a course**. In this use case, a student requests to register a course. The system then checks whether the student's prerequisites are satisfied. If so, the system will let the student register and record the registration. Otherwise, the system notifies the student that he cannot register by this system.

1. Identify the classes used in this use case. Use UML diagram to give the names of the classes, the attributes and operations for each classes. You just need to give the attributes and operations related to this use case. Then give the associations and multiplicities of the associations.

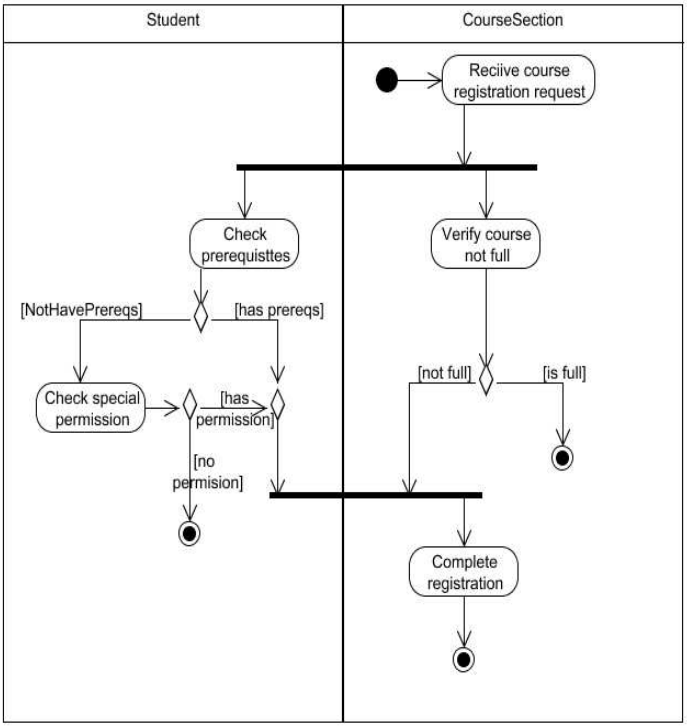
2. Draw the sequence diagram to show the process of register a course. In this diagram an object of GUI is added.

3. For a planned course section, if not enough students are registered, then the course section will be cancelled. If the number of registered students reaches the maximum size, then the course section will be closed. Use a State diagram to describe the different states of the course section.

4. Design a user interface for this sub-system.

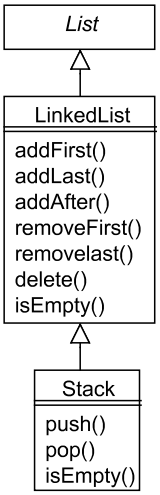
Problem 4 (10 marks)

Write down the information you get from the following activity diagram. Explain what are `student` , `CourseSection`, `Check prerequisites`; what are the meanings of other symbols; etc. (if space in this page is not enough, then you can continue on the back of this page or back of the previous page.)



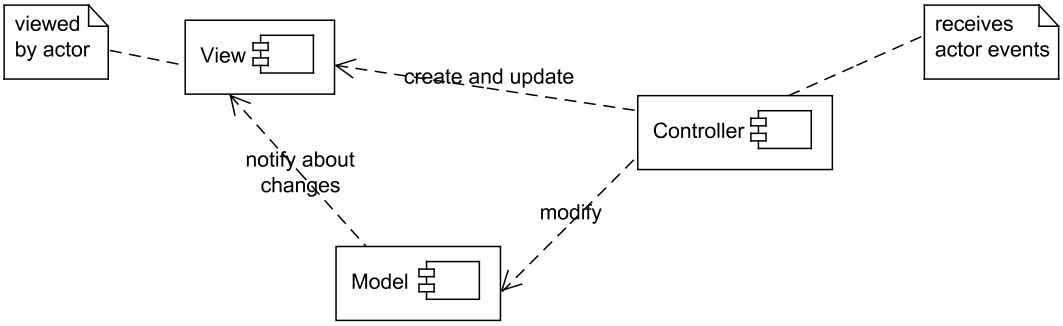
Problem 5 (10 marks)

The following diagram is for generalization of some classes. Indicate why this design is not very good and give a better design.



Problem 6 (10 marks)

The following diagram explains the Model-View-Controller(MVC) architectural pattern.



1. Explain what kinds of objects are contained in the instance of three components: `Model, View` and `Controller`.

2. Explain why MVC architectural pattern is good for increasing cohesion and reducing coupling.

Problem 7 (10 marks)

Read the following Java source codes.

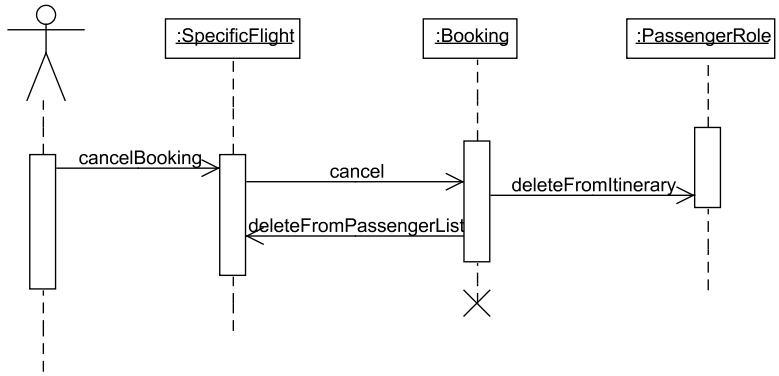
```
public routineX(String command)
{
  if(command.equals("drawCircle")
    {
     drawCircle();
    }
   else
    {
      drawRctangle();
     }
}
```

1. Why the program is not very good?

2. Suggest methods to improve the program.

Problem 8 (10 marks)

The following sequence diagram describes a use case: **cancel booking** in an air-ticket booking system.



1. Explain the meanings of the UML symbols used in this diagram.

2. Draw a class diagram for the use case. You need to draw classes with operations and associations with multiplicity.