

CS 3415 FINAL EXAMINATION

December 13th, 2007

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. Aggregations are special associations in UML class diagrams. ()
2. Object Constraint Language is a formal language designed to enhance the modeling capabilities of UML. ()
3. In UML, an association between two classes describes a relationship between these classes in a class diagram. ()
4. A system design is based on system requirements analysis. Therefore the requirements cannot be changed after the design started. ()
5. In UML notation, a class is represented as a box which contains name, attributes, operations and generalizations. ()
6. A good user interface should let a user always knows what to do next. ()
7. A good design pattern should be an efficient solution for a specific problem. ()
8. A sequence diagram shows the sequence of messages exchanged by the set of objects performing a certain task. ()
9. A subsystem is part of a large system, and which has a definite interface. ()
10. In bottom-up testing, you start by testing only the user interface, with underlying functionality simulated by stubs. ()

Problem 2 (20 marks)

Answer following questions:

1. What is a black-box testing and what is a glass-box (white-box) testing?

2. What are the advantages and disadvantages of black-box and glass-box testing?

3. Give an example where using a black-box testing is more suitable.

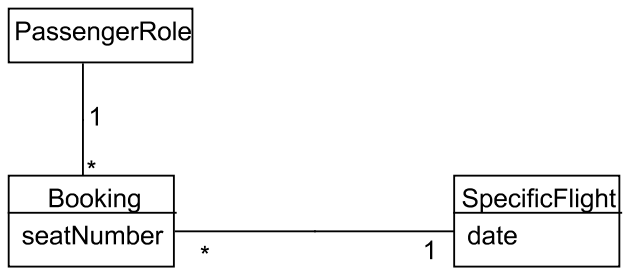
4. After you have a flow graph of a subsystem, how do you design a glass-box testing?

Problem 3 (25 marks)

The TB Airlines runs sightseeing flight from Thunder Bay. The company runs several daily numbered flight on a regular schedule. Now they want to develop an Airline Reservation System (ARS). The system keeps track of passengers who will be flying in specific seats on various flights, as well as people who will form the crew. For the crew, the system needs to track what everyone does, and who supervises whom.

1. There are several classes identified: `PersonRole`, `RegularFlight`, `Booking`, `SpecificFlight`. Identify other classes for ARS. Also identify necessary attributes and operations for all the classes (include identified classes and the classes you defined).

2. Draw the class diagram for the ARS which should include the following parts.



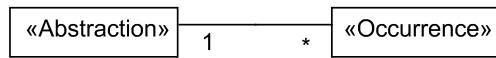
3. Develop a use case: Book a ticket by using a sequence diagram. You need to use instance of classes `PassengerRole`, `Booking`, `SpecificFlight` shown previously. But in previous diagram, no operations are given in the classes. So you should first add the operations to the classes.

4. Use a state diagram to describe the behavior of the booking system.

5. Design a user interface for this system.

Problem 4 (10 marks)

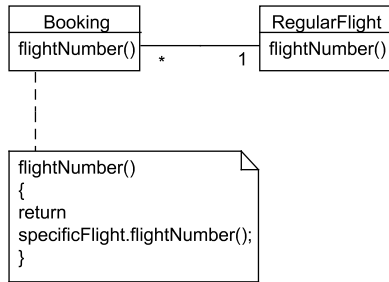
The following design pattern is called Abstraction-Occurrence pattern which can be used to avoid duplications in class diagram.



Give an example to use this pattern. Explain what are the advantages of using this pattern in your example.

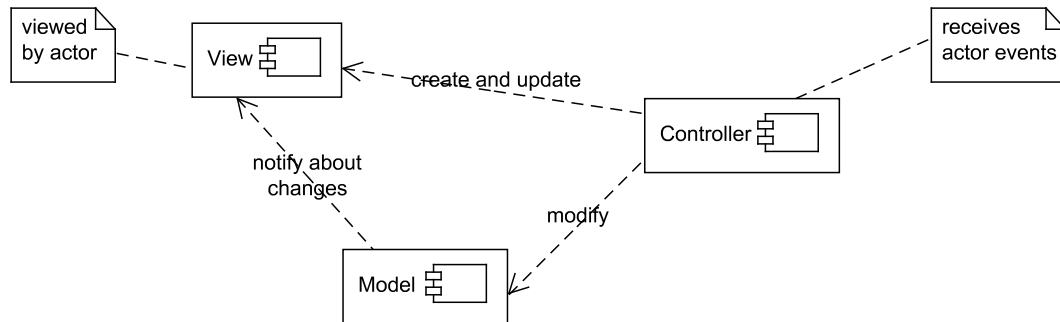
Problem 5 (10 marks)

The following diagram is part of an air flight booking system. 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.

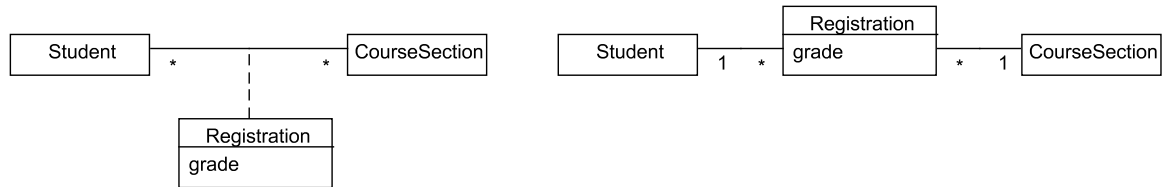


1. Explain what kinds of objects are contained in the instance of three components: Model, View and Controller.

2. Give an example which uses this MVC architecture.

Problem 7 (15 marks)

The following two diagrams are used for requirement documents of a Course Register System.



1. Are these diagrams describes a same thing? If they are different, then indicate the differences. If they are same, explain the associations among the three classes.

2. Can we combine the classes `Registration` and `CourseSection` into one class so that the diagram is more abstract? Why?

3. Draw a linked instances diagram generated from the right class diagram.