University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

SEMESTER 1 EXAMINATION – 2009/2010

COMP 30160

Object-Oriented Design

Prof. V. Callaghan

Prof. J. Carthy

Dr. Mel Ó Cinnéide*

Time Allowed: 2 hours

Instructions for candidates

Answer Question 1 and two other questions.

Question 1 (compulsory)

Answer each of the following questions. In each case, only a short answer is required.

(i) Distinguish between *Iterative* Development and *Incremental* Development.

(ii) Explain what is meant by the *architecture* of a software system.

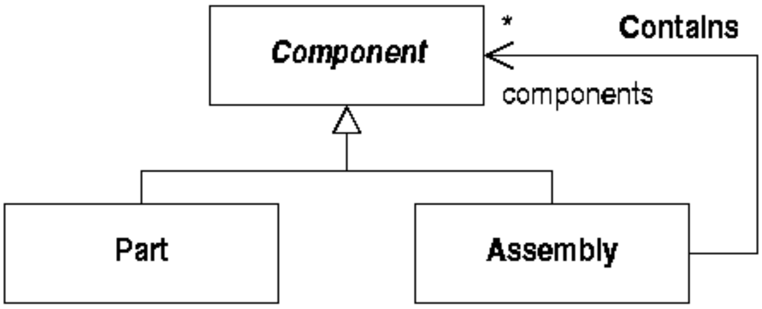
(iii) Explain the slogan *You Ain't Gonna Need It* in the context of Agile Processes.

(iv) How are testing and refactoring related in the context of Extreme Programming?

(v) What are the negative consequences of decomposing a Use Case model extensively using *includes* and *extends* relationships?

(vi) In modelling with UML, what is the key goal in building Interaction Diagrams? Which models may change as result of this process?

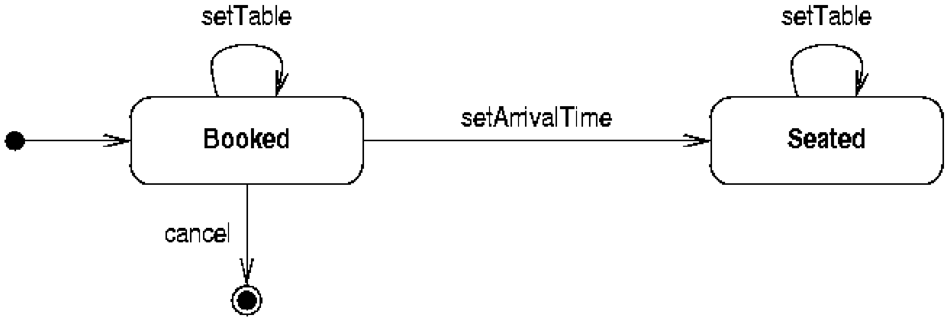
(vii) What type of object structure is denoted by the following class diagram? Why might aggregation be more appropriate?



(viii) Classes `A` and `B` implement the interface `Inf`. It transpires that a new method must be added that makes sense for `A` but not for `B`. Explain briefly the possible solutions and their pros and cons.

(ix) What is an *association class*? Under what circumstances should an association class be promoted to a proper class?

(x) Explain in words the operation of the following Statechart.



(xi) Explain the Lack of Cohesion in Methods (LCOM1) metric from the Chidamber&Kemerer metrics suite.

(xii) Why is is recommended that a superclass should not be a concrete class?

(xiii) What is multiple dispatch? Provide an example where double dispatch might be useful.

(xiv) The Visitor pattern may require breaking the encapsulation of the visited objects. Why?

(xv) Explain how the Factory Method pattern can be used to connect parallel class hierarchies.

(30 marks, 2 marks per part)

Question 2

(a) Describe the Waterfall Process. What are the drawbacks of this process? Describe Extreme Programming in detail, explaining how it addresses the drawbacks of the Waterfall process.

(12 marks)

(b) Describe how to do Test-Driven Development using JUnit. What are the benefits and drawbacks of this approach?

(8 marks)

Question 3

(a) (i) For each of the four UML diagram types *use case diagram*, *class diagram*, *sequence diagram* and *statechart*, describe the essential notation briefly and explain what the diagram is used for in modelling a software system. Describe how the different diagram types are related to one another.

(8 marks)

(ii) For each of the diagram types, explain how it can be used to drive the the actual implementation of the software system.

(4 marks)

(b) Draw a UML class diagram that models the following facts about a library. Explain carefully the design decisions you take, as well as any limitations of your model (please note that this discussion is an vital part of your answer).

> For each book held by the library, the catalogue contains the title, author’s name and ISBN number of the book. There may of course be multiple copies of a book in the library. Each copy of a book has a unique accession number. There are many registered readers belonging to the library, each of whom is issued with a number of tickets. The system records the name and address of each reader, and the number of tickets with which they have been issued. Readers can borrow one book for each ticket that they possess, and the system keeps a record of which books a reader has borrowed, along with the date by which the book must be returned.

(8 marks)

Question 4

(a) (i) What is the intent of the Observer pattern? Draw the typical class diagram for this pattern and show, using a sequence diagram, a typical scenario for this pattern.

(6 marks)

(ii) In relation to the Observer pattern, answer the following:

a. In what way can this pattern lead to unexpected updates?

b. How can the observation of multiple subjects be handled?

c. How can the deletion of subjects and/or observers be handled?

d. Explain the push and pull models.

e. Explain the options for *notification*.

f. How can you ensure that subject state is consistent before notification when subject subclass operations invoke inherited state-changing operations?

(6 marks)

(b) A programmer introduces the Abstract Factory pattern to a program. Explain carefully the effect that this will have on each of the Chidamber and Kemerer metric values for this program, i.e., for each metric, will it become better or worse and in what way? What conclusion do you come to regarding the value of the Chidamber and Kemerer metrics suite?

(8 marks)