S228/209

DUBLIN INSTITUTE OF TECHNOLOGY

DT228 BSc. (Honours) Degree in Computer Science

Year 2

SUMMER EXAMINATIONS 2015/2016

SOFTWARE ENGINEERING II [CMPU2020]

MR RICHARD. LAWLOR

WEDNESDAY 18TH MAY 4.00 P.M. – 6.00 P.M

TWO HOURS

ATTEMPT FOUR OUT OF FIVE QUESTIONS
ALL QUESTIONS CARRY EQUAL MARKS

1. (a) Discuss the pros and cons of class inheritance versus interface inheritance in building flexible and loosely coupled reusable software. Provide a simple diagram with explanations to illustrate the advantages of using an interface.

(12 marks)

(b) What are design patterns and what are their purpose?

Describe the Adapter design pattern and elaborate on the differences between class adapter and object adapter.

(13 marks)

2. You are required to do some object-oriented design for a standalone restaurant software system that mainly manages bookings. The restaurant software should be able to handle advance reservations, walk-in bookings, assigning tables to reservations and so on.

(a) A layered architecture allows for separation of concerns. Explain what is meant by this. Then describe an appropriate layered architecture for the restaurant system given that it will be implemented as standalone software.

(6 marks)

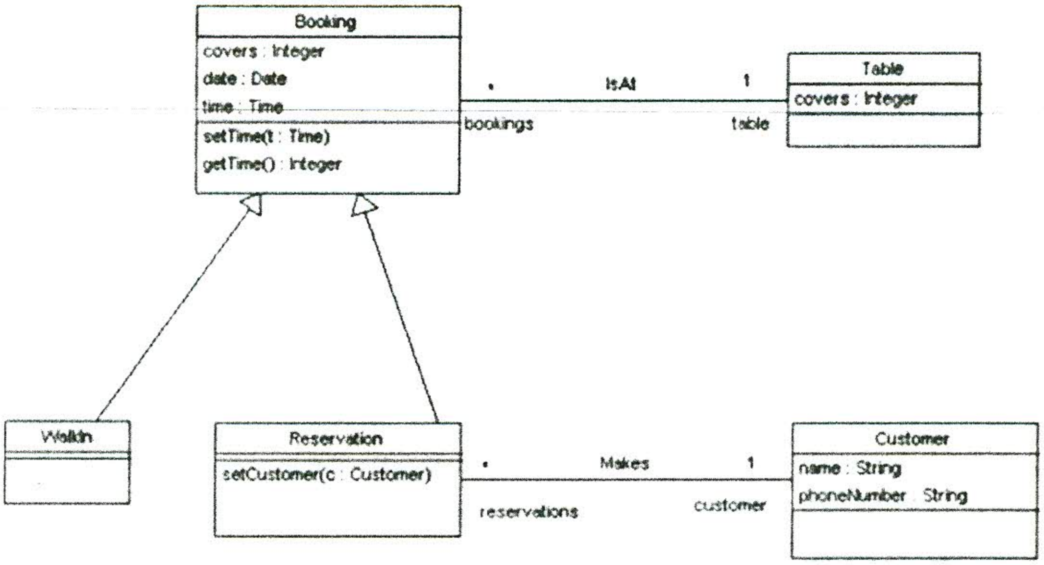
(b) Describe the structure and purpose of the Observer design pattern and in which generic situations it might be applicable.

(7 marks)

(c) Show with the aid of a package/class diagram and a sequence diagram, and explain in writing how the Observer pattern could be incorporated into the design of the restaurant system to couple application code with user interface code.

(12 marks)

3. The following class diagram is an incomplete UML specification for a restaurant booking system. OCL will be used to specify business rules and in order to test the specification it has been decided to model it in USE (UML Specification Environment).



(a) One business rule to be added to the specification is: "bookings for the same table must not overlap". Express this in OCL. You can assume that a booking requires 2 hours and that **getTime()** returns the starting time of a booking in minutes.

(10 marks)

(b) Provide USE code to define the classes **Booking** and **Reservation**. Also attempt to provide SOIL implementations for the operations **getTime()** and **setCustomer()**.

(10 marks)

(c) Suppose an extra operation for Booking is declared:

assignTable(t : Table)

Write an OCL precondition for this operation that represents the constraint: "the table must be large enough to seat all the people in a booking".

(5 marks)

4. (a) Explain what is meant by *Design by Contract* (DbC).

(8 marks)

(b) Comment on DbC from the following viewpoints:

i) Precision versus Detail
ii) DbC and Quality Assurance

(9 marks)

(c) Write an appropriate contract in Spec# for a method `int ISqrt(int x)` that computes the integer square root of x.

Supposing you then implemented the method in C#, what could be done with the Spec# contract?

(8 marks)

5. (a) Describe six of the key practices of the agile methodology XP.

(12 marks)

(b) Discuss *Essential versus Accidental Complexity* in software design.

(13 marks)