Ollscoil na hEireann
THE NATIONAL UNIVERSITY OF IRELAND

UNIVERSITY COLLEGE DUBLIN

CHRISTMAS EXAMINATIONS, 2007

SCBDF0005/SCBDF0015 - B.Sc. HONOURS DEGREE EXAMINATION
ARBDF0015 - B.A. (COMPUTER SCIENCE) HONOURS DEGREE EXAMINATION

COMPUTER SCIENCE

30160: Object-Oriented Design

Dr. Paul McCullagh
Dr. Joe Carthy
Dr. Mel Ó Cinnéide*

Answer any *four* questions. **All questions carry 20 marks.**
(Time allowed: 2 hours)

1. Consider the following scenario. You are a software engineer with several years experience of working on projects that used the Waterfall model for software development. Six months ago you started work with a new company that uses Extreme Programming (XP) as its development methodology.

Write an account of your experiences in this new environment, comparing and contrasting your present experiences of XP with your previous work.

Your account should demonstrate that you understand the two methodologies mentioned above, and that you appreciate the differences between them. You are encouraged to describe fictitious events and experiences to enhance your account. You may adopt any position you wish, pro-XP or anti-XP or somewhere in between, so long as your position is supported by informed argument.

(20 marks)

2.

(a) Draw a UML class diagram for the Observer pattern. Draw a UML sequence diagram that depicts the interactions that occur after the subject is updated.

(10 marks)

(b) Describe the 3-tier layered architecture. What are the benefits of this architecture? Explain clearly the role the Observer design pattern can play in this architecture.

(10 marks)

3.

(a) Describe in detail any one of the Gamma *et al* design patterns with which you are familiar, **excluding Oberver**. Include the intent, applicability and structure of this pattern. Describe also the consequences and implementation issues related to this pattern that you find interesting

(10 marks)

(b) In relation to **any two** design patterns, describe them briefly, provide a motivating example of how they might be used together and explain carefully the synergy between them.
(A full description of these patterns is not required. You may use the pattern described in (a) if you wish, but it is not required.)

(10 marks)

4.

(a) Explain the purpose of each of the three principal models constructed when analysing using UML, and explain how these models relate to each other. What aspects of a system *cannot* be described using these models?

(10 marks)

(b) Explain briefly how each of the following types of association can be implemented in an object-oriented programming language of your choice. Discuss any implementation issues that arise.

(i) Unidirectional one-to-many association
(ii) Bidirectional one-to-one association
(iii) Bidirectional many-to-many association
(iv) Association class

(10 marks)

5.

(a) What is a framework? Explain the *inversion of control* principle in this context. Distinguish between *template*, *abstract* and *hook* methods in a framework.

(10 marks)

(b) Describe the Open-Closed principle. Give an example of a class that is closed, describe how it may be made open, and list the advantages of that accrue from making it open.

(10 marks)

6.

(a) Consider the following class hierarchy:

```
class A{
    public A(string s){name=s;}
    public int foo(){…}
    private string name;
}

class B extends A{
    public B(){super("qwerty");}
    public void foobar(){…}
}
```

The programmer has implemented B as a subclass of A, but later realises that it is more appropriate to model this relationship as a composition. Rewrite the class B to perform this refactoring. Do not change A, and all clients of B should continue to work as before without being updated. Comment on the nature and purpose of this type of transformation.

(10 marks)

(b) The following program excerpt provides a simple example of a violation of a well-known object-oriented design heuristic. What is the heuristic being violated? Explain in detail what problems can occur when this heuristic is violated. How would you refactor this code to solve the problem?

```
class Employee{
    …
    public int foo(){
        …
        … = mgr.getDept().getName();
        …
    }
    private Manager mgr;
}

class Manager {
    …
    public Department getDept(){
        return managedDepartment;
    }
    private Department managedDepartment;
}

class Department{
    …
    String getName(){
        return name;
    }
    private String name;
}
```

(10 marks)