University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

SEMESTER 2 EXAMINATION – 2013/2014

COMP 30160

Object-Oriented Design

Prof. Alain Mille

Mr. J. Dunnion

Dr. Mel Ó Cinnéide*

Time Allowed: 2 hours

Instructions for candidates

Answer Question 1 and any two other questions.

All questions carry equal marks (20).

Question 1 (compulsory)

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

(i) Outline the main problems with the Waterfall methodology. What are the key features of current software methodologies?
(ii) Explain the roles of testing and refactoring in the context of Extreme Programming.
(iii) Explain what is meant by *object identity*. What impact this has on an *association class* in the UML class model?
(iv) Show how you might implement an aggregation relationship with a cardinality of `1` to `1..*` in any object-oriented language with which you are familiar.
(v) Classes `A` and `B` both 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 the advantages and disadvantages of each solution.
(vi) Describe the Response For a Class (RFC) metric from the Chidamber&Kemerer metrics suite and explain what a low or high value of this metric can mean.
(vii) In the context of testing, define *equivalence class partitioning* and *boundary value analysis* and explain what these techniques are used for.
(viii) Explain briefly the Law of Demeter, and the benefits of following it.
(ix) Explain covariance and contravariance and demonstrate how the latter is necessary to ensure type security.
(x) Explain the role of design patterns in software development.

(20 marks, 2 marks per part)

Question 2

(a) Describe the Unified Process and contrast it with an Agile Process such as Extreme Programming.

(10 marks)

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

(10 marks)

Question 3

(a) Draw a Control Flow Graph (CFG) for the following method. Provide three sets of test cases, each providing a different level of code coverage, namely (i) statement coverage, (ii) modified condition/decision coverage and (iii) path coverage.

```
boolean foo(int i, int j){
        if (i<=0 || j>100)
                System.out.println("...");
        while(i>0 && j<100){
                i=i/2;
                j++;
        }
        return j==100;
}
```

(10 marks)

(b) Describe the *Open/Closed Principle* the *Dependency Inversion Principle*. Provide an example of a module that is both open and closed and explain in what way it is open and in what way it is closed. Show also how a direct dependency between classes can be inverted, again explaining your example.

(10 marks)

Question 4

(a) In relation to any **two** design patterns with which you are familiar, answer the following. Take care to answer parts (i) to (iii) for each design pattern in turn.

(i) What is the intent of the pattern?

(ii) Using the appropriate UML diagrams, describe the typical structure and interactions of the pattern.

(iii) Describe two issues related to the applicability or implementation of this pattern that you find interesting.

(10 marks)

(b) For each of the four code smells listed below,

- describe the code smell
- suggest how this smell can come about in practice
- outline the refactorings you might perform in order to remove the smell.

(i) Feature Envy

(ii) Speculative Generality

(iii) Duplicated Code

(iv) Refused Bequest

(10 marks)