



University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

Spring, 23/24 TRIMESTER EXAMINATIONS

COMP47480

Contemporary Software Dev

Module Coordinator: Assoc Professor Mel Ó Cinnéide

Student Number

--	--	--	--	--	--	--	--

Seat Number

--	--	--	--

Time Allowed: 120 minutes

Materials Permitted in the Exam Venue:

English language dictionary (hard copy)
Foreign language dictionary (hard copy)
Non-programmable or scientific calculator

Materials to be Supplied to Students:

12 Page Answer Booklets

Instructions to Students:

Attempt all questions. Question 1 is in 10 parts, (a) to (j).

Question 1

(a) What in your opinion are the three most important advantages of an Agile development process over a BDUF/Waterfall approach?

(b) "A student is characterised by a name and a student number, while a module is characterised by a code and a name. The same module can be delivered several times, and each delivery of a module is associated with a year and a term. A student enrolls in several deliveries of modules, some of which may be repeat attempts of the same module. Each enrolment is associated with a grade and the number of lectures the student attended."

Create a UML Class Diagram to model the above description. No need for methods or field types. Show classes, fields and relationships between classes, including cardinalities.

(c) Draw a Control Flow Graph (CFG) for the following method. Construct a truth table to determine a minimal set of test cases that yield Modified Condition/Decision Coverage for the `while` loop only.

```
boolean foo(int i, int j, int k) {  
1   if (i == j)  
2       System.out.println("equality");  
3   while((i > 0 && j < 100) || i == k){  
4       i = i / 2;  
5       j++;  
6   }  
7   return j == 100;  
8 }
```

(d) Explain what cyclomatic complexity is, and write a note on the pros and cons of its use as a complexity metric.

(e) In relation to the SOLID principles SRP, LSP and ISP, write a note indicating the impact the principle has on coupling and cohesion.

(f) (i) Provide a code sketch of a class that observes the OCP, and explain clearly in what way it is closed and in what way it is open. (ii) How should the OCP be applied in Agile development?

(g) You encounter two fragments of duplicated code in a code base. (i) What factors determine whether or not they should be refactored? (ii) Assuming they should be refactored, what strategies might you use?

(h) In relation to overengineering, describe (i) how it comes about (ii) what problems it causes and (iii) how it may be avoided.

(i) What are the two main approaches for a Strategy to access its Context class, and what are the pros and cons of each approach?

(j) In relation to the State pattern, what are the possibilities for the location of the state-transition code, and what are the pros and cons of each approach?

(70 marks)

Question 2

Write a short essay (~2 pages) on any aspect of internal software quality that you find particularly interesting. Relevant topics include software principles, refactoring, code smells, metrics, design patterns, or any combination of these topics. You may cover several topics, or do a "deep dive" into a narrow area.

Advice: This is a chance to show how you have engaged with the material of the module. If possible, choose an area you have given some thought to, and have experience of outside the module (work, internship, additional reading etc.). Aim for quality rather than quantity.

(30 marks)

oOo