

EXAMINATIONS — 2007
MID-YEAR

COMP462
OBJECT-ORIENTED
PARADIGMS

Time Allowed: 3 Hours

Instructions:

- *Read each question carefully before attempting it.*
- This examination will be marked out of **180** marks.
- Answer all six questions. Each question has the same value, and should take approximately 30 minutes to answer.
- You may answer the questions in any order. Make sure you clearly identify the question you are answering.
- Many of the questions require you to discuss an issue, or to express and justify an opinion. For such questions, the assessment will take into account the *evidence* you present and any *insight* you demonstrate.
- Some of the questions ask for examples from object-oriented languages. Your answers need only refer to object-oriented languages discussed in the course, but you may refer to other languages if you wish.
- Non-electronic foreign language-English dictionaries are permitted.

Question 1.

[30 marks]

You are working on a project using RUP and UML. Two of your fellow developers — Una and Cho — are having an argument about how to communicate their design to the rest of the team. Una insists that they should write UML class diagrams and sequence diagrams, while Cho says they should use CRC cards and focus on responsibility. Choose a side and give arguments for it, plus arguments against the opposition.

Question 2.

[30 marks]

You are the coach of an XP project. Your boss — the project manager — asks you what metrics she should use to a) determine the project's progress, and b) to keep track of code quality. What do you propose, and why?

Question 3.

[30 marks]

What is the difference between a framework and a library? What things would you do differently when designing classes that will be part of a framework versus designing classes that will be part of a library? How do the principles of object-orientation support libraries and frameworks?

Question 4.

[30 marks]

You have been asked to write the **design guidelines** for your next project. This project will use a programming language, like Eiffel or C++, that supports Multiple Inheritance. What guidelines will you provide for using this feature? Would you forbid its use in any circumstances, allow multiple inheritance in particular cases, or encourage its general use?

List your design guidelines for Multiple Inheritance in this new language, justify each guideline choice, and explain how the guidelines will help you avoid a Big Ball of Mud.

Question 5.

[30 marks]

Eiffel, C++, and now Java and C# all support generic types and classes (i.e parametric polymorphism). Describe how you could add support for Aspect Orientation to such a language. You should discuss whether aspects can be applied to generic types and generic classes; whether or not aspects can be made generic; and how this would affect the type system and the language implementation.

Question 6.

[30 marks]

Describe how you could use meta-programming to support the implementation of design patterns. Do you think that using a prototype-based language would make it easy?
