TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

EXAMINATIONS — 2009

MID-YEAR

COMP462
OBJECT-ORIENTED PARADIGMS

Time Allowed: 3 Hours

Instructions:

- This examination will be marked out of **180** marks.
- Read each question carefully before attempting it.
- 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. Extreme Subtyping [30 marks]

(a) [10 marks] Briefly describe three practices that are used to design classes in extreme programming.

(b) [10 marks] Give an example where a subclass should **not** be considered as a subtype? In general, explain why (or why not) all subclasses should be subtypes.

(c) [10 marks] Which of subtyping or subclassing is more important to Extreme Programming? Explain your answer.

Question 2. System Design [30 marks]

"System Design? — that's for people who are afraid of rolling in big balls of mud!"

(Attrib. to Thomas J. "Tad" Peckish.)

(a) [10 marks] Will CRC cards help you to "roll in the big balls of mud" of your programs? Why or why not?

(b) [20 marks] In *No Silver Bullet*, Fred Brooks lists the following technologies as "hopes for the sliver bullet":

i) object-orientation

ii) graphical programming (visual languages)

iii) program verification (mathematical proof techniques for programs)

iv) programming environments and tools (IDEs)

For each of these, explain the benefits they provide, and why they have turned out not to be silver bullets.

Question 3. Measuring Multiple Inheritance [30 marks]

(a) [10 marks] Imagine you re-wrote a Java program to use Multiple Inheritance. Explain how and why this would change the value of the DIT (depth of inheritance tree), NOC (number of children), and WMC (weighted methods per class) metrics?

(b) [10 marks] Assume you had a programming language with multiple dispatch. Explain when you should use multiple dispatch in your program. Show some example code that does not use multiple dispatch, and show how you could rewrite that code to use multiple dispatch.

(c) [10 marks] Imagine you tried to design a programming language with both multiple dispatch and multiple inheritance. Explain why this is harder than designing a language with only one of these features. Would you keep both in your language design? If so, explain why. If not, explain which one you would keep, and why.

Question 4. Aspects and Reflection [30 marks]

(a) [10 marks]

Explain how aspects can be regarded as part of both the meta-level and the base-level of a programming language.

(b) [20 marks]

Describe the advantages and disadvantages of implementing aspects using reflection.

Question 5. Frameworks and Ownership [30 marks]

Discuss the similarities and differences between frameworks and libraries. How are these differences reflected in the ownership structures of frameworks and libraries?

Question 6. Java [30 marks]

All quotations in this section are taken from Gilad "Winston Smith" Bracha's weblog, *Room 101*, `gbracha.blogspot.com`, post titled *Original Sin*, Wed May 27 2009. Gilad Bracha was the Java lead at Sun Microsystems (now Oracle Corporation) and was primarily responsible for getting Generics into Java.

(a) [10 marks]

> *I've often said that Java's original sin was not being a pure object oriented (sic) language — a language where everything is an object.*

List five ways in which Java is **not** a pure object-oriented language. Do you think impurity is Java's "original sin"? Why or why not?

(b) [10 marks]

> *If int is a subtype of Object ([in contrast to Java]) we'd expect int[] to be a subtype of Object[], because in Java we expect covariant subtyping among arrays. Of course that isn't type safe, . . . I've come to the conclusion that Java made the right call on array covariance in the first place.*

Describe why covariant subtyping causes problems for arrays in Java, and generics in Eiffel. Explain why Java generics do not have this problem. Do you agree with Gilad when he says that Java's covariant arrays are the right design choice? Why or why not?

(c) [10 marks]

> *One of the many reasons I left Sun was because it was increasingly impossible to make meaningful progress on the language, or anything else for that matter.*

Imagine that you were hired to replace Gilad's replacement at Oracle Corporation. Propose a change to Java that you think would "make meaningful progress on the language", and justify your proposal.

****