TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI

VICTORIA
UNIVERSITY OF WELLINGTON

**EXAMINATIONS — 2011**

MID-YEAR

# SWEN423

# Object-Oriented Paradigms

**Time Allowed:** Three 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. [30 marks]

One day you wake up and suddenly find yourself as a software architect consultant for a large telecommunications company. You have a meeting with a project manager in 30 minutes where you will need to outline the choices available for the programming languages to use when designing the software solution. The Telco is keen on using the latest (and not always greatest) technologies available and thus you will have to adopt an agile (preferably SCRUM) development approach. To impress the manager, you will not use run of the mill Java/ANYTHING.NET as your languages of choice and you feel pity on their mental capacity and thus decide to avoid functional programming or logic programming paradigms. You also cannot use decades old technology such as Self and Smalltalk since this will go against the motto of the company: ″Latest IS Best″.

*This means that you need to quickly introduce the manager to the beauty of Newspeak and DeepJava, outline the benefit of using either in the agile setting, and propose your personal choice of the language including how it can help you with creating a sensible architecture and design for your product solution.*

To make your task more concrete, here is the exciting product you will be working on (surely the only such product in development in Wellington!): it is a web site for internal use that will have a database behind it that can support both report production and data entry by its users. The most exciting project you can ever dream of working on.

You now only have 28 minutes left, GO!

## Question 2. [30 marks]

**(a)** [10 marks] Draw an instance diagram including `java.lang.Class`, `java.lang.Object`, and `java.lang.String` objects with the ″instance of″, ″subclass of″, and ″class″ relationships clearly shown. Write a short justification for each relationship.

**(b)** [10 marks] Please explain how *mirrors* help provide metaclass-like facility in object-based languages such as JavaScript.

**(c)** [10 marks] Identify a scenario where it would be advantageous to use mirrors or metaclasses. Clearly state and justify whether you would prefer to use mirrors or use metaclasses in your scenario.

## Question 3. [30 marks]

Discuss how *metrics* can be used to help evaluate *aspect-oriented* programs in real life. Suggest some metrics that you think would be useful (or obvious but NOT useful) and discuss why (or why not) they would (or would not) be useful.

## Question 4. [30 marks]

Please clearly state the meaning of and differences between as well as state and justify your opinion of:

1. Frameworks
2. Patterns
3. Libraries

Feel free to use examples of libraries or patterns or frameworks to justify your points!

## Question 5. [30 marks]

**(a)** [10 marks] What is *model driven engineering*?

**(b)** [20 marks] How can model driven engineering help avoid "Big Balls of Mud" in software development?

## Question 6. [30 marks]

**(a)** [15 marks]

Imagine that you are a lecturer tasked with teaching "system design for object-oriented systems". Outline the design and the implementation of a tool that you would want to develop to assist you when teaching system design.

If you like, you can assume that this is going to be used in SWEN301 or similar course where the students are familiar with programming in the small but are yet to understand the concept of systems in the large. If you think it is better to teach system design to novice programmers, you can take this route too as long as you justify your choice.

Justify the selection of concepts you would prefer to teach and what kinds of techniques would be essential to demonstrate when using such tool with your students.

**(b)** [15 marks]

Please tell us why *Object-Oriented Programming is better than Functional Programming*. You may not agree with this, but in such case simply make the Devil's Advocate argument.

********************************