

END-YEAR

COMP 462

OBJECT-ORIENTED PARADIGMS

Time Allowed: 3 Hours

Instructions:

- *Read each question carefully before attempting it.*
- This examination will be marked out of **120** marks.
- 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 express and justify an opinion. For such questions, you will be assessed on your *justification*.
- Many 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.
- **Note:** There are two parts to this exam, Part 1 and Part 2. Please use separate answer booklets for each of the two sections, and clearly label answer booklets either Part 1 or Part 2.

Part 1

Question 1. Inheritance

[20 marks]

Discuss what inheritance means in the context of object-oriented programming languages, that is, discuss the fundamental concepts that most forms of inheritance provide in one form or another.

You should *not* just produce a list of the different kinds of inheritance that are available in existing languages. Your discussion should include descriptions of the fundamental concepts and an explanation as to why they are useful. You may use examples to support your description from any object-oriented language you are familiar with (not just those discussed in the course).

Question 2. Short answer questions

[40 marks]

Discuss any **four** of the following topics (each worth 10 marks).

(a) [10 marks] Explain the similarities and differences between *delegation*, such as that used in Self, and *inheritance*, such as that used in Smalltalk.

(b) [10 marks] Polymorphism in object-oriented languages generally means that the actual code that is executed is determined at run-time. Discuss the different ways that polymorphism is provided in object-oriented languages.

(c) [10 marks] In Smalltalk, all classes fit into one hierarchy with the class `Object` at the top, whereas C++ has many small hierarchies. Compare and contrast the advantages and disadvantages of the different approaches. Your answer may refer to approaches taken by other languages.

(d) [10 marks] A common argument against providing multiple inheritance is that it introduces ambiguity, since the same name may be used by more than one ancestor class. Discuss the different ways that languages that have multiple inheritance deal with the ambiguity problem.

(e) [10 marks] The concept of *mixin* is sometimes described as providing abstract *sub* classes. Explain, with examples, what this means. Your examples can be in any language, including non-existent ones.

(f) [10 marks] Java does not have multiple inheritance, but it does have the **interface** mechanism. Explain how this mechanism can be used to provide one of the benefits of multiple inheritance.

Part 2

Question 3. Design Processes

[15 marks]

How does use of CRC cards relate to Wirfs-Brock's "Responsibility Driven Design" process? Discuss how they work together, and how they conflict with each other.

Question 4. Evaluation

[15 marks]

Explain the role of heuristics and metrics in object-oriented design, and discuss their limitations.

Question 5. Frameworks and Patterns

[15 marks]

Explain the difference between object-oriented frameworks and object-oriented Design Patterns, and describe the role of each in object-oriented design.

Question 6. Design and Frameworks

[15 marks]

Discuss how the development an object-oriented design in general relates to the development of an object-oriented framework. In particular, consider whether every object-oriented design should involve a framework.

(Recall the views of Johnson in *Frameworks = Components + Patterns*, and the views of Schmid in *Systematic Framework Design by Generalization*.)
