EXAMINATIONS — 2002

MID-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.
- Answer all questions.
- 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, you will be assessed on your answer, the *evidence* you present, and any *insight* based on these.
- 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. Object-Orientation [20 marks]

Consider the following two sayings of Bertrand Meyer regarding Object-Oriented analysis:

(a). *"... objects are just there for the picking"*. 1989.

(b). *"[in finding objects] we should not expect too much"*. 1999.

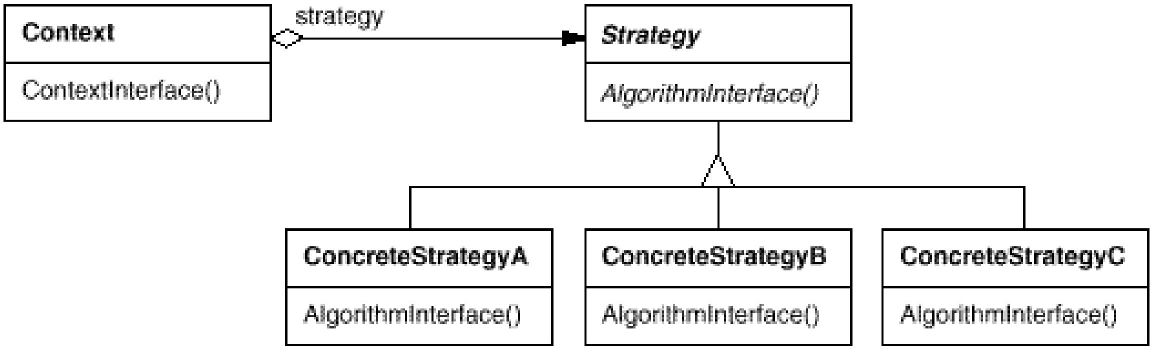
Discuss how easy it is to identify objects and classes that lead to good object-oriented designs, with reference to use cases, essential use cases, CRC cards, and Responsibility-Driven Design.

Question 2. Development [20 marks]

Compare and contrast Responsibility-Driven Development (RDD) and Extreme Programming (XP) as techniques that lead to good object-oriented designs. Which do you think is more likely to lead to a Big Ball of Mud? Why?

Question 3. Design [20 marks]

The following figure shows the solution of the Strategy pattern:



(a). Patterns are supposed to improve or simplify designs. Discuss the Strategy pattern with respect to one of Riels' heuristics and one of Chidamber and Kemerer's metrics.

(b). Would this pattern help in the design of an object-oriented framework? If so, how? If not, why not?

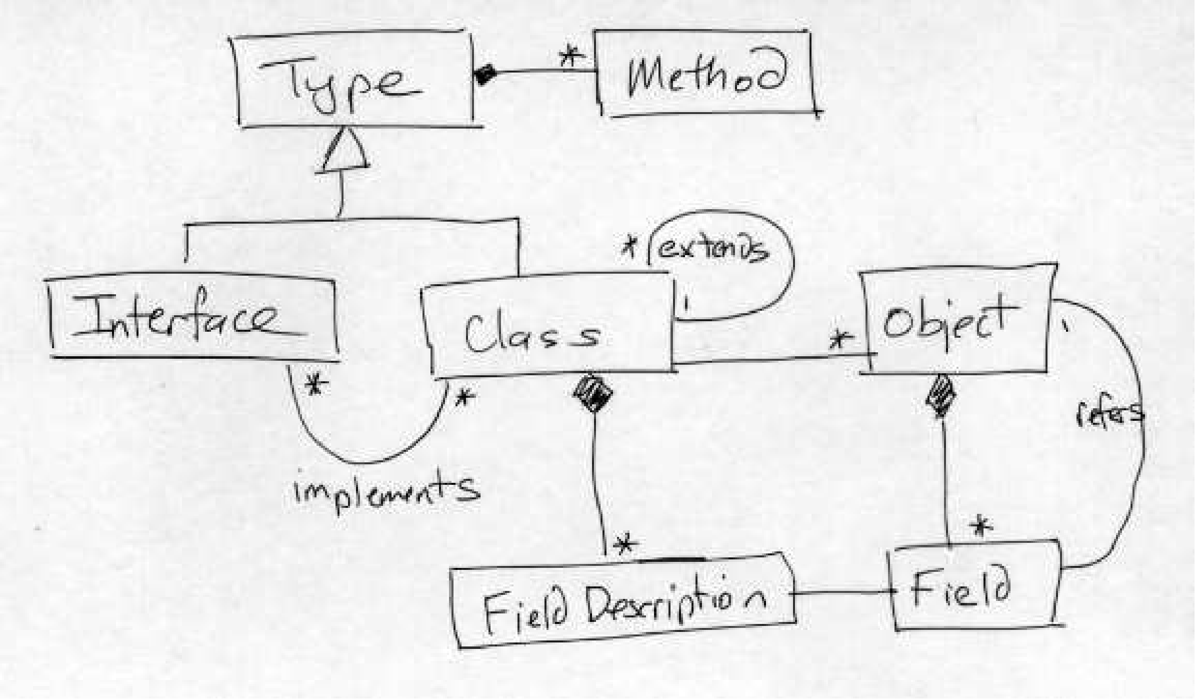
Question 4. Inheritance [20 marks]

(a). In Java, 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.

(b). How does this choice affect the design of generic collection class libraries?

Question 5. Metamodels [20 marks]

The following UML diagram shows some of the runtime structure of a Java-like language:



(a). Draw a similar diagram for a Self-like prototype-based language.

(b). Use the diagrams to explain two advantages and two disadvantages of prototype-based languages compared to Java-like languages.

Question 6. Object-Oriented Paradigms [20 marks]

> *Object-Orientation is dead — it has run its course.*
> *Aspect-Orientation is the new paradigm,*
> *and will completely replace Object-Orientation*
> *the way Object-Orientation replaced Structured Programming,*
> *or the way night follows day.*

Discuss.

****