

**CS\_M71 2003/04**  
**Design Patterns and Generic Programming**

*(Attempt 2 questions out of 3)*

**Question 1** Constructing, destructing and copying objects

(a) Special member functions

- (i) What are the “special member functions” of a class (in C++), and what is special about them regarding declaration and definition? [3]
- (ii) Discuss the meaning of each of the special member functions, and try to give reasons why the language emphasizes them. [4]
- (iii) When do objects get destructed? [2]

**[9 marks]**

(b) RAII

- (i) Explain the “Resource Acquisition Is Initialisation” idiom, and discuss its importance. [4]
- (ii) Describe in general terms how RAII is used in the following applications:
  - 1. connecting to a database;
  - 2. writing to a file;
  - 3. producing recursively nested structures like “{([)]}” or as they occur with XML or HTML applications.

**[3]**

**[7 marks]**

(c) Copying objects

- (i) Explain what is meant by “deep copying” and “shallow copying”. [2]
- (ii) Give an example where we create a “dangling pointer” by shallow copying. [2]
- (iii) Explain the idea of “Copy on Write” (COW). [2]
- (iv) Explain why the assignment operation is especially critical with respect to “exception safety”. [3]

**[9 marks]**

**Question 2** Class hierarchies

(a) Access control

- (i) Which access specifiers exist in C++ and what are their meanings? Explain in general terms which members should be public and which should be private or protected. [4]
- (ii) Explain the differences between public, protected and private inheritance. When should public inheritance be used, and when should protected or private inheritance be used? [4]

[8 marks]

(b) Member resolution

- (i) Explain how the static and how the dynamic type of an object is determined, and under what circumstances the static or the dynamic type of an object is used. [4]
- (ii) Discuss the meaning and usage of abstract classes. [4]

[8 marks]

(c) Design

- (i) Discuss in general terms the assumptions on the relations between a public base class B and a derived class D. Give examples when deriving D from B can be misleading (you might use the “Square vs. Rectangle” discussion). [5]
- (ii) Describe and discuss the “Template Design Pattern”. [4]

[9 marks]

**Question 3** Polymorphism and generic programming

(a) Forms of polymorphism

- (i) Explain the three forms of polymorphism we discussed in the lecture, and give examples for each case. [6]
- (ii) Discuss the advantages and disadvantages of using templates and using polymorphic classes (in C++). [4]

[10 marks]

(b) Freestanding functions

- (i) Discuss when to use member functions and when to use freestanding functions (in C++). Explain how “argument-dependent name lookup” (ADL) helps to establish the “Interface Principle” (IP). [5]
- (ii) Explain how “tagging classes” are used to avoid a combinatorial explosion of concepts, and how we can instrument global functions to be aware of this kind of “tagging polymorphism”. [5]

[10 marks]

(c) Generic programming

- (i) Explain what a “concept” is, and why it is important. [5]

[5 marks]