UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

MASTER OF SCIENCE IN COMPUTER SCIENCE

PROGRAM AND ALGORITHM DESIGN

Thursday 27 April 2000

2:00 to 3:30

Examiners:

I. Watson (External)

R.N. Ibbett (Chairman)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. (a) C++ allows access controls to be specified for class members.

   i. What is the most common access control used for instance variables?

   ii. How about for member functions?

   iii. What is the implication of this for users of the class?

   iv. And for implementors? *[6 marks]*

(b) Two ways of achieving the reuse of code from some class A in a new class are by having B inherit from A and by having an instance variable of class A in class B.

   i. Give two reasons for preferring inheritance.

   ii. Give one reason for preferring the second approach. *[6 marks]*

(c) Consider the following declaration of a class member function:

```
void fred(A * a) { ... }
```

Modify this declaration to indicate that `fred` does not change either the object it is called on or the object passed to it as argument `a`. *[3 marks]*

(d) Describe a common situation when it is advisable to define a destructor function for a class. *[5 marks]*

(e) i. What's the difference in C++ between a *conversion* and a *cast*?

   ii. From a programming methodology point of view, why is it useful to have conversions in a language and why also might they be considered undesirable? *[5 marks]*

2. (a) Algorithms can commonly be described in both an iterative and a recursive style. Describe one advantage of each approach. *[4 marks]*

(b) Why would a vector from the C++ STL (Standard Template Library) not be a good datastructure for storing the characters in the text buffer of a text editor? What would be a better datastructure? *[3 marks]*

(c) What two primary operations are more efficient on a *deque* than a *vector*? *[2 marks]*

(d) Say variable `x` is of type `vector<int>`. Write a `for` loop that uses an iterator to set a previously declared local variable `n` of type `int` to the sum of the values stored in `x`. *[5 marks]*

(e) Give tight big-O bounds on the worst-case time complexities of the following operations:

i. finding if a given item is in a sorted STL vector,

ii. adding a new item to an STL map object,

iii. deleting an arbitrary item from an unsorted STL list. *[3 marks]*

(f) Consider the following heap-ordered tree.

```
          9
        /   \
      7       5
    /  \     /
   3    2   4
```

i. Describe how the above tree would typically be stored.

ii. Draw the new state of the tree after a `delete-max()` operation is performed.

iii. Draw the new state of the tree if *instead* an `insert(6)` operation is performed. *[8 marks]*

3. Below is the start of a description of a set of classes for a diary program that might run on a personal digital assistant.

```
class Time ; // stores both date and time.

class Person {
private:
  string name ;
  string tel_number ;
...
}

class Entry { // An entry in the diary
private:
  Time from ;
  Time to ;
public:
  bool check_overlap(Time begin, Time end) ; // returns true if overlap with
                                             // stored time.
  void display() ;
...
};

class Lecture : public Entry {
private:
  string subject ;
...
}

class Meeting : public Entry {
private:
  list<Person> attendees ;
public:
  add_person(Person p) ;
...
}
class Diary {
private:
  Container<Entry> entries ; // see below for discussion of class Container.
public:
  add_entry(Entry & e) ;
  display_entries(Time begin, Time end) ;
  ...
}
```

QUESTION CONTINUES ON NEXT PAGE

(a) Draw a UML-like class diagram showing the static relationships between the above classes. Include associations with their navigabilities and multiplicities, and generalization relationships. When listing member functions and instance variables of classes, omit type and argument information unless particularly relevant (it might be useful to show some instances of the `Time` type, for example). [*10 marks*]

(b) The container class `Container` in the above code is unspecified. If fast operation of the `display_entries` member function is desired, what kind of datastructure would be appropriate? [*2 marks*]

(c) i. Which class or classes would it make sense to have *abstract*? Why?

ii. Which member functions if any in each of these abstract classes would it make sense to have *pure* and which not? Why? [*5 marks*]

(d) i. What does it mean to say that an association is *compositional*?

ii. Indicate which associations in your diagram it would be sensible to think of as being compositional. [*6 marks*]

(e) In class we briefly discussed software architectures and control models. If this diary were to be connected to a clock and expected to sound reminders about entries starting soon, what kind of control model might be appropriate? [*2 marks*]