

(MID-YEAR)

COMP 462

Object-Oriented Paradigms

Time Allowed: THREE HOURS

Instructions:

- *Read each question carefully before attempting it.*
- This examination will be marked out of 180 marks. The marks are specified in square brackets “[ ]”. The marks for parts of questions are specified in parentheses “( )”.
- You must answer each of the “top-level” questions however you have some choice within some of these questions. Do not answer more than is asked — *the extra answers will be ignored*. Cross out any solutions you do not wish to be considered.
- 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*.

1.

[60 marks]

Consider building an application to manage entries in a bibliography. A bibliography consists of a set of *entries*, each entry consists of a unique label used to identify the entry, a “kind” (for example, book, journal article, technical report), and a list of *fields*. Each field consists of the name of the field (for example, author, title, publication date) and its value. The management of these entries includes reading and parsing a text form of the entries and allowing the user to perform various operations (add entry, list entries, and so on) specified from an interface, and then write the results out in the text form.

This question looks at various aspects of an object-oriented design of this application. You must evaluate a number of design decisions. If you feel there is not enough information to make a decision, describe what is missing and how it affects your answer.

(a)

(25 marks)

It has been suggested that the way to start an object-oriented design is to consider the *nouns* and *verbs* that appear in the statement of the requirements: the nouns are the classes and the verbs are the operations. This technique will not give a complete solution. For example not all nouns necessarily make good classes and some required operations may not be explicitly stated.

For each of the classes below, write a paragraph justifying whether or not it is appropriate for this application:

- i. a class representing the database,
- ii. a class representing each field,
- iii. a class representing the interface,
- iv. a class representing parsing the text file, and
- v. a class representing the value of the field.

(b)

(12 marks)

Part of an object-oriented design is describing the relationships between the classes. Decide whether the following (independent) design decisions are appropriate for this application and justify your decision.

- i. There are two classes. One is `Entry`, which represents one entry in the database, and the other is `Field`, which represents one field in an entry. `Entry` is derived from (a subclass of) `Field` so that `Entry` can have access to the private state of `Field` without `Field`'s encapsulation being compromised by other classes.
- ii. There are a number of classes. One, `Entry`, is an “abstract” class representing entries in the database. The rest are “concrete” classes which represent each particular kind of bibliography entry (classes `Article`, `Journal`, `TechReport` and so on). Each concrete class is derived from (a subclass of) `Entry`.

- (c) (10 marks)
- One of the criteria in object-oriented design is that of hiding implementation details from those parts of the application that do not need to know about them.
- Suppose there are three classes, one representing the database, one representing an entry in the database, and one representing a field in an entry. Based on the criteria stated above, it seems reasonable to require that the implementation of the database does not know about the existence of the field class. Is it reasonable? Justify your answer.

- (d) (13 marks)
- A “map” class (also known as “dictionary” and “table”) maintains a relationship between a unique “key” and its corresponding “value”. Since the label for each entry is unique, one possible design is to represent the database as a map between labels and entries. Discuss the issues associated with this design.

2. [20 marks]  
 Object-oriented programming is about “objects”. Discuss what is generally meant by “object”. Your discussion should include such things as what an object looks like, what properties it has, how one identifies what objects should appear in a system, and how objects are used to implement a system. You should use the terminology commonly used when discussing objects.

3. [20 marks]  
 In *Object-oriented analysis and design*, Booch defines encapsulation as follows:

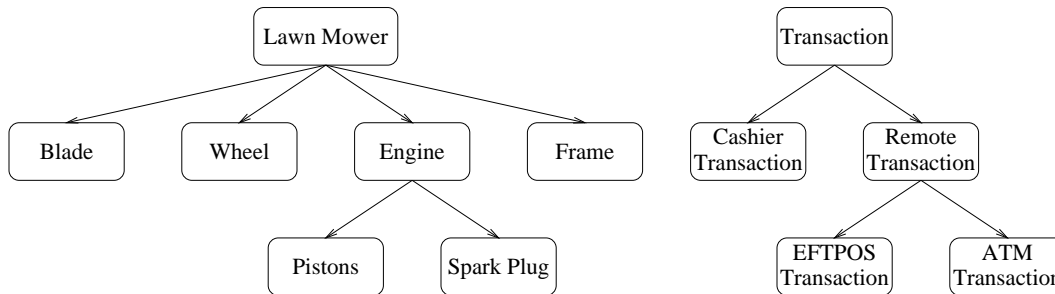
*Encapsulation is the process of compartmentalizing the elements of an abstraction that constitute its structure and behaviour; encapsulation serves to separate the contractual interface of an abstraction and its implementation.*

Compare and contrast the facilities for supporting encapsulation of any **two** of C++, Smalltalk, CLOS, Eiffel, Ada, and Emerald. Your discussion should include (but not be limited to) reference to Booch’s definition.

4. [20 marks]  
 This question is about *inheritance*, a feature that is considered to be an important part of object-oriented programming by many (in fact some say it is *the* important part).

Answer any **two** of the following questions.

(a) [10 marks]  
 Inheritance is usually used to represent “hierarchy”. Below are two hierarchies. Explain which is best represented by inheritance and why.



(b) [10 marks]  
 In fact there are two views as to what inheritance means. Compare and contrast these two views.

(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.

5.

[20 marks]

This question is about *polymorphism*, a feature that is considered to be an important part of object-oriented programming by many (in fact some say it is *the* important part).

Answer **two** of the following questions about polymorphism.

(a) (10 marks)

In most object-oriented languages, there is a strong relationship between inheritance and polymorphism. Explain what this relationship is.

(b) (10 marks)

Polymorphism in C++ is done by specifying functions as *virtual* whereas in Smalltalk no such specification is required (essentially, all functions are virtual). Why did C++'s designer, Bjarne Stroustrup, provide polymorphism this way?

(c) (10 marks)

Assume that B is a class that has an equality operator `operator==` with the usual meaning and no virtual functions and assume that D is derived from B. Consider the following declarations, where the bodies of the two functions are exactly the same, except for the pointer dereferencing required.

```
B dostuffwith(B in_parameter);  
B* dostuffwith(B* in_parameter);
```

```
B aB;  
B* pB; // Initialised with the same value as aB  
D aD;  
D* pD; // Initialised with the same value as aD
```

For the following code fragments, say whether the fragment is legal, in which case you should describe what happens and why, otherwise explain why it is illegal.

```
i. aD = dostuffwith(aD);  
   pD = dostuffwith(pD);  
   if (aD == *pD)  
       cout << "They are the same\n";  
   else  
       cout << "They are different\n";
```

```
ii. aB = dostuffwith(aD);  
    pB = dostuffwith(pD);  
    if (aB == *pB)  
        cout << "They are the same\n";  
    else  
        cout << "They are different\n";
```

6. [20 marks]

Answer any **two** of the following questions.

(a) (10 marks)

Discuss what is meant by “reference-semantics” in the context of C++. You should explain what it means, how C++ supports it, and how well C++ supports it. You should also mention any issues relating to providing it in a language.

(b) (10 marks)

Discuss the importance of *object identity* with respect to providing *persistence* in an object-oriented language.

(c) (10 marks)

Some object-oriented database management systems (OODBMS) provide so-called “complex values”. Explain what these are, whether or not they fit into the traditional object model, and why it is appropriate that they appear in OODBMS.

(d) (10 marks)

Object-oriented database management systems (OODBMS) are essentially database management systems that support the object model. Explain why this is a reasonable thing to want to do and how the use of the object model in OODBMS seems to differ from its use in object-oriented programming languages.

7. [20 marks]

Answer any **two** of the following questions

(a) (10 marks)

Discuss the phrase “message passing”. Why is it reasonable to use this to describe communication in an object-oriented programming language? In what way might it be misleading?

(b) (10 marks)

In Emerald, types are only related *implicitly*, that is, the relationship is not specifically stated such as it might be in a language that has inheritance. Explain why Emerald made this choice. In particular, discuss the advantages

and disadvantages of this choice.

- (c) (10 marks)  
What are the issues associated with including *concurrency* in the Object Model? Your answer should include examples describing how this has been done from at least two languages.
- (d) (10 marks)  
Explain, with an example, how the ability to *overload operators* can make code more reusable.

\*\*\*\*\*