U08186 Advanced Object-Oriented Programming

Examination Rubric

Examination length: **2 hours.**

Answer **three** questions.

Section A is compulsory and contains one question, worth 40 marks.

Two further questions must be answered from Section B. Each is worth 30 marks.

The total number of marks is 100.

Examination Questions

Section A - Compulsory

Question A1

a) Draw the class diagram for the Strategy design pattern, explaining the meaning of the aggregation symbol, the inheritance arrow and the interface label applied to the class Strategy. Your explanations must be in the particular context of that design pattern.

10 marks

b) Write the code for an Iterator class for the class `ArrayList` and show how you would add an `iterator` method to the `ArrayList` class that creates and returns an iterator object of that class.

10 marks

c) Which design pattern do you think is being described here in an extract from a design patterns textbook. Explain your answer.

> "Then she took me to the set of documentation. I am not making this up: there were 8 feet of manuals for me to read ... each page was $8\frac{1}{2} \times 11$ inches and in small print! This was one complex system! Now, if you and I and say another four of five people were on a project that needed to use this system, not all of us would have to learn the entire thing. Rather than waste everyone's time, we would probably draw straws, and the *loser* would have to write routines that the rest of us would use to interface with the system"

4 marks

d) What are the similarities and differences between the Adapter pattern and the Facade pattern in terms of what they do and the circumstances in which they are typically used?

6 marks

e) The Command design pattern can be used to implement the response to menu item and button clicks on a GUI for Microsoft Word, for example. Show how this may be achieved by drawing a class diagram of the pattern instantiated to this particular situation. Explain the advantage of using the Command pattern in this situation.

10 marks

Section B - Answer two questions

Question B1

a) Java 7 contains a syntax for binary literals. Illustrate this by writing code that assigns a 32-bit number of your choice to an integer variable `i`. The number should be broken up into 8-bit chunks for ease of reading.

5 marks

b) Explain the syntax and semantics of two language features new to Java 7 that, if present, would tell you that a `try ... catch` block was written to be compiled by a Java 7 compiler rather than a Java 6 compiler.

5 marks

c) Consider the following signature for a method designed to print all the elements of a list.

```
void printAll(List<?> list);
```

What are the advantages of making the parameter be `List<?>` rather than `ArrayList<?>`?

4 marks

d) One might expect the parameter to have the type `List<Object>`. That way, it seems likely, you could call it with parameters of type `List<String>` or `List<E>` for any class `E`. Explain carefully what is wrong with this argument and why the type `List<?>` is used instead.

6 marks

e) In Java 7, it is possible to write `ArrayList<>`, for example, with no type specified for the elements. Give an example of a situation in which this is allowed and explain why it is allowed?

5 marks

f) Suppose that for some interface type `I`, the method signature was rewritten as follows:

```
void printAll(List<? extends I> list);
```

Explain, without simply reusing the word “extends”, what property this new signature now enforces upon the parameters with which the method `printAll` is called. Include in your answer a possible situation in which this restriction will be useful by suggesting a possible interface `I`.

5 marks

Question B2

a) Explain what the Spec# precondition:

```
requires forall{int i in (0: a.Length),
  int j in (i: a.Length); a[i] <= a[j]};
```

says about the array `a`.

3 marks

b) An informal specification for a method states that it should accept an array of integers `a` (indexed from zero) and an integer `x` and return the position of `x` in the array. Explain why this specification is inadequate.

2 marks

c) Write a specification, including a contract in the form of pre- and post-conditions in the manner of Programming by Contract, which is fragile and which allows the programmer of the method to make an assumption. You may express your contract in mathematical notation or in the syntax of Spec# or in very clear English.

4 marks

d) Write a different specification, also in the form of pre- and post-conditions, which is defensive and does not allow an assumption but returns a special value in a certain case. You may express your contract in mathematical notation or in the syntax of Spec# or in very clear English.

3 marks

e) Explain how having a specification in the form of a contract simplifies the work of a programmer of a method.

3 marks

f) Explain how having a specification in the form of a contract simplifies the work of a user of a method.

3 marks

g) Write an `assert` statement in Java to test the precondition in part a) and print out a message if it is not satisfied. You may find it helpful to define the precondition itself in a separate boolean-valued method and call it in the `assert` statement itself.

6 marks

h) Suppose a Java method called `find` implements the specification in c). Write a test using one of the methods of the JUnit framework to check that the postcondition of `find` is true in a situation where the precondition holds.

6 marks

Question B3

a) Consider the following program fragment:

```
int daysInMonth(int year, int month) {
switch(month) {
    case 2: if(leap(year)) days = 29; else days = 28;
    case 4: case 6: case 9: case11: days = 30;
    default: days = 31;
   }
   return days;
}
```

Assuming that the above fragment is written in Java, explain what would happen when a programmer tried to compile and run a Java program containing the method `daysInMonth` and the statement:

```
n = daysInMonth(2011, 4);
```

4 marks

b) Assuming that the above fragment is written in C#, explain what would happen when a programmer tried to compile and run a C# program containing the method `daysInMonth` and the statement:

```
n = daysInMonth(2011, 4);
```

4 marks

c) The following fragment of C# includes the declarations of some properties:

```
public class Customer
{
     private int numberOfOrders = 0;

    public int ID
     {
         get
         {
             return numberOfOrders;
```

```
            }
            set
            {
                numberOfOrders = value;
            }
        }
    }
```

Suppose that a variable customer of class `Customer` has been declared, instantiated and initialised. Write statement(s) in C#, using the properties, that will increase the customer's number of orders by 1.

4 marks

d) Write the statement(s) necessary to increase the customer's number of orders by 1 using "getter" and "setter" methods, named `getNumberOfOrders` and `setNumberofOrders`, as conventional in Java. Assume that these methods have been declared and defined.

4 marks

e) The .NET framework allows parts written in a variety of programming languages to be composed in a single program. Describe a possible circumstance in which this might be desirable.

2 marks

f) Explain the role of the Common Intermediate Language (CIL) in making this possible.

4 marks

g) Likewise, explain the role of the Common Type System (CTS) in making this possible.

4 marks

h) A programmer running programs using the .NET framework notices a delay when running programs for the first time but no delay on subsequent runs. Explain why this is so.

4 marks

End of Examination Paper