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) A shop makes four different types of cake, each with a different price:

1. sponge cake, priced at 199p

2. sponge cake with icing, priced at 299p

3. sponge cake with icing and sprinkles, priced at 349p

4. sponge cake with sprinkles, priced at 249p

Consider the interface `Priceable` defined below:

```
interface Priceable {
    int getPrice();
}
```

You need to write a class `Cake` that implements `Priceable` and calculates the price of a cake.

i) Write the class using the Strategy Pattern. Note that this will involve writing a number of auxiliary classes. Some of them will be very similar to each other. You do not need to write them all out provided that you make the pattern clear. There is no need to define constructor methods for the classes.

8 marks

ii) Write the class using the Decorator pattern. Again, there is no need to define constructor classes.

8 marks

iii) Draw the class diagram for the Decorator pattern, and explain in detail why your solution conforms to the Decorator pattern. You may do this using the Strategy pattern instead if you are more confident with your solution to the first part.

6 marks

iv) explain why the solution using Decorator pattern is better than the solution that uses the Strategy pattern.

6 marks

b) The State pattern makes it possible for an object to appear to change its class. Explain how this happens.

6 marks

c) The Singleton pattern ensures that only one instance of a class is created. Explain how it does this.

6 marks

Section B - Answer two questions

Question B1

a) Explain, with an example in each case, how the use of generics and enumerated types in Java programs reduces the possibility for run-time error.

8 marks

b) Write a method that uses an enhanced for loop to concatenate together the elements of an ArrayList of Strings supplied as a parameter.

5 marks

c) Explain why the enhanced for loop is considered preferable to the for loop in this case.

2 marks

d) Explain the difference in meaning between the following five uses of the <> notations with reference, if necessary, to where in a program you may encounter them.

i) `ArrayList<Int>`

3 marks

ii) `ArrayList<E>`

3 marks

iii) `ArrayList<?>`

3 marks

iv) `ArrayList<? extends I>`

3 marks

v) `ArrayList<>`

3 marks

Question B2

a) Explain what the precondition of this Spec# method means. Note use of "!":

```
void High (int !a[], out int high) {
requires
    a.Length != 0;
ensures
    exists {int i in (0..a.Length): a[i] == high} &&
        forall{int i in (0: a.Length); a[i] <= high};
// code goes here
}
```

5 marks

b) Explain the meaning of the postcondition of the Spec# method in part a). Note that ranges in Spec# are "half-open": `i in (m: n)` means that $\mathtt{m} \geq \mathtt{i} < \mathtt{n}$.

5 marks

c) A programmer implements the method from part a in this way.

```
high = -int.MaxValue;
int j = 0;
while (j != a.Length) {
    if (a[j] > high) high = a[j];
    j = j + 1;
}
```

Comment on this and reason about whether it satisfies the specification.

5 marks

d) Would the above implementation satisfy the specification if the line

```
requires
    a.Length != 0;
```

had been omitted? Explain.

5 marks

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

5 marks

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

5 marks

Question B3

a) The following fragment of C# includes the declarations of "properties".

```
public class C {
    private static int size;
    public static int Size {
        get { return size; }
        set { size = value; }
    }
}
```

Write statement(s) in C# using the properties that will set size to 3.

4 marks

b) Write statement(s) in C# using the properties that will increase size by 1.

4 marks

c) Write the statement(s) necessary for part b above using conventional "getter" and "setter" methods called `getSize` and `setSize`. Assume that these methods have been declared.

4 marks

d) A programmer wishes to use a large scientific program written in FORTRAN but via a graphics user interface which is to be written in C#. Explain the role of the Common Intermediate Language (CIL) of the .NET framework in making this possible.

5 marks

e) Explain the role of the Common Type System (CTS) in making this possible.

4 marks

f) 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.

4 marks

g) A C# compiler will report an error when trying to compile the following. Explain why that is, why reporting an error is a good idea and how the program fragment can be corrected.

```
int DaysInMonth(int year, int month) {
    switch(month) {
        case 2: if(leap(year)) days = 29; else days = 28;
        case 4: case 6: case 9: case 11: days = 30;
        default: days = 31;
    }
    return days;
}
```

5 marks

End of Examination Paper