

## 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 diagram of the Template Method pattern, including the use of a note to indicate the implementation of an operation where necessary. You do not need to remember the official Gang of Four names for the classes and operations you write but any alternative names you choose should reflect the purpose of these roles and classes. Indicate abstract classes and operations clearly either by using italics or by using the property list {abstract}.

Remember that you can use the forward slash notation thus: /abstractSomething/ to indicate abstract classes and operations.

**4 marks**

- b) What is the Hollywood Principle and where is it being used in this pattern?

**4 marks**

- c) The Facade pattern reduces coupling between client and subsystem classes. Explain what this means, how it happens, and why it is useful.

**6 marks**

- d) The State pattern appears to cause the object that represents the state to change its class. Explain how it does this.

**4 marks**

- e) The rest of this question is about the Decorator design pattern.

- i) Draw the diagram for the Decorator design pattern, naming the classes Component, Decorator, ConcreteDecoratorA, ConcreteDecoratorB and ConcreteComponent.

**4 marks**

- ii) Imagine that a pizza company serves many different kinds of pizza base, each of which can have one or more kinds of topping. Example pizzas include Pepperoni Deep Pan and Pineapple Thin 'n' Crispy with Extra Cheese. The choice of base and the choice of topping(s) both affect the prize that the customer must pay. Imagine you are asked to write a program that calculates the prize of a pizza using the Decorator pattern. Give the bindings you would use to instantiate the classes and operations of the Decorator pattern to write the program.

**5 marks**

- iii) Assuming the existence of appropriate constructors, give the code necessary to create a Pineapple Thin 'n' Crispy with Extra Cheese and ask for its price.

**3 marks**

- iv) The use of polymorphism in the Decorator pattern illustrates the Open-Closed Principle. Explain what this principle means with reference to the Decorator pattern.

**4 marks**

- v) It is possible, though not advisable, to implement pizza pricing in a procedural manner with just one class, ie not using the Decorator pattern and not using polymorphism either. What would the attributes of this class be and how would the price of the pizza be calculated from it?

**6 marks**

## Section B - Answer two questions

### Question B1

- a) Write an definition in Java of an enumerated type that can be used to represent the four different states of a set of traffic lights.

**4 marks**

- b) Write code that can be used to simulate enumerated types.

**4 marks**

- c) Write the code for declaring a light and initialising it to red, for each of the two different implementations.

**4 marks**

- d) Imagine that you wish to pass an Array List as a parameter to a method `doSomething` but you want `doSomething` to be flexible enough to accept Array Lists of Strings but also Array Lists of Integers. Assuming that `doSomething` does not return a value and does not take another parameter, give its declaration and say what aspect of generics you are using.

**4 marks**

- e) What feature of Java 5.0 would be used to access an Array List of Integers as though it contained ints?

**2 marks**

- f) Give the code for `doSomething` assuming that its purpose is to print out all the elements of the Array List. Give three different forms of the code as follows:

i) one using the enhanced for loop,

**3 marks**

ii) one using the `Iterator` interface explicitly

**3 marks**

iii) one indexing the `ArrayList` with an integer.

**4 marks**

g) What is the relationship between these three different solutions?

**2 marks**

**Question B2**

Consider the implementation of a stack as the class `Stack` in C# supplied in the appendix.

- a) Imagine you have asked Microsoft Visual Studio, or some other IDE, to produce unit tests and the following method is produced.

```
public void popTest()
{
    Stack_Accessor target = new Stack_Accessor();
    // TODO: Initialize to an appropriate value
    int expected = 0; // TODO: Initialize to an appropriate value
    int actual;
    actual = target.pop();
    Assert.AreEqual(expected, actual);
    Assert.Inconclusive
        ("Verify the correctness of this test method.");
}
```

Explain what this code does and why the IDE produces it.

**5 marks**

- b) Explain why the IDE produced the second assertion.

**3 marks**

- c) Similarly, imagine that the following code has also been produced.

```
public void pushTest()
{
    Stack_Accessor target = new Stack_Accessor();
    // TODO: Initialize to an appropriate value
    int elem = 0; // TODO: Initialize to an appropriate value
```

```
target.push(elem);  
Assert.Inconclusive  
    ("A method that does not return a value cannot be verified.");  
}
```

Given the message of the second assertion, how should the method `push` be tested?

**4 marks**

d) What are the advantages of unit testing compared to manual testing?

**3 marks**

e) What are the advantages of using a language like Spec# in place of or in conjunction with unit testing?

**3 marks**

f) Give four examples of where Spec# annotations could be useful for this class. Include accompanying reasons and syntax where applicable.

**12 marks**

**Question B3**

Answer the following questions about C#.

- a) While maintaining a C# program, you encounter the following code:

```
struct Date
{
    public int day, month, year;
    public Date(int day, int month, int year) {
        this.day = day;
        this.month = month;
        this.year = year;
    }
}
```

What is the difference between making Date be a struct and making it be a class.  
Explain the advantage of using a struct in this case.

**5 marks**

- b) Explain what properties are and how they are used.

**5 marks**

- c) Define properties for one of the attributes of Date.

**5 marks**

- d) Redefine these as read-only auto-implemented properties.

**5 marks**

- e) Consider the following code.



```
int [,] arrs;  
arrs = {{1,2},{3,4,5}};
```

i) This code uses rectangular arrays. Explain how you know this and what it means

**2 marks**

ii) Why does the code not compile?

**2 marks**

iii) Change the code so that it compiles without changing what is to be stored in the array.

**6 marks**

**End of Examination Paper**

## Appendix A Implementation of a Stack in C#

```
class Stack
{
    int[] elems;
    int size;
    static int SIZE = 10;

    Stack()
    {
        elems = new int[SIZE];
        size = 0;
    }

    int pop()
    {
        size--;
        return elems[size+1];
    }

    void push(int elem)
    {
        elems[size] = elem;
        size++;
    }
}
```