ISYS1083/1084 Object Oriented Software Design

TuteLab 5 - Creational Patterns

Question 1

a) What is the difference between the Factory Method and AbstractFactory patterns?
b) When should a Factory Method be encapsulated in a separate class?
c) Give two examples where the Factory Method pattern is suitable
d) Give two examples where the AbstractFactory pattern is suitable

Question 2 - Think about how you would write an XML parser that works for both constructing a document object model (DOM) and SAX events so it can be piped into another XML processor. What design pattern will you use?

(a) Draw the UML class diagram for your proposed solution.

(b) Sketch out a UML sequence diagram for your solution showing both generic and specific objects as it reads the following XML :

```
<designer name="Martin Odersky">
    <language name="Pizza">
    <language name="Scala">
</designer>
```

Question 3 - An accounting application to compute company tax is to be developed. It will be used in all of the states of Australia and include GST, asset tax, carbon tax, capital gains tax, etc. The rates of each tax vary across states and some states don't implement some tax types at all.

Suggest an appropriate pattern giving two reasons. Explain briefly how this pattern helps compute the total company tax effectively for different states.

Additional Self-Guided Study Scenario

A barcode generator is required for the ACME food manufacturer. Every barcode number must be unique, following the format “ACME-xxxx-xxxx” and never reused for different products.

Question 4

a) What is the main purpose of the singleton pattern?
b) Why is the constructor made private in a Singleton pattern?
c) Give one advantage of a Singleton over a global variable?
d) What will be the output of the program below?

```
class X {
   private static X uniqueInstance;
   static int count = 0;
   private X() { count++; }
   public static X getInstance()
   {  if (uniqueInstance == null)
          uniqueInstance = new X();
```

```
      return uniqueInstance;
   }
}
public class TestX {
   public static void main(String args[])
   {  X s1 = X.getInstance();
      X s2 = X.getInstance();
      System.out.println("Number of Xs = " + X.count);
   }
}
```

1. Number of Xs = 0
2. Number of Xs = 1
3. Number of Xs = 2
4. None of the above