Uva Wellassa University, Sri Lanka
Faculty of Science & Technology
Computer Science & Technology Degree Program
1st Semester Examination February/March 2012
CST 422-2 Software Design using Design Patterns

Uva Wellassa University

Time Allowed: **Two (02) hours** Number of Questions: Five (05)

Number of Pages: Six (06)

Instructions:

Answer **all** questions

Q1.

Choose the most suitable design pattern which covers the following design principals and explain your answer briefly

a) "**The Hollywood Principle**" - Don't call us, we will call you
   i. Template pattern
   ii. State pattern
   iii. Factory pattern
   iv. Strategy pattern

b) "**The Dependency Inversion principle**" Depend upon Abstractions. Do not depend on concrete classes
   i. Template pattern
   ii. State pattern
   iii. Factory pattern
   iv. Strategy pattern

c) "**Program to an interface not to an implementation**"
   i. Strategy pattern
   ii. Observer pattern
   iii. Command pattern
   iv. Iterator pattern

d) "**Strive for loosely coupled designs between objects that interact**"
   i. Strategy pattern
   ii. Observer pattern
   iii. Command pattern
   iv. Iterator pattern

e) "Favor **composition over inheritance**"

i. Strategy pattern
ii. Observer pattern
iii. Command pattern
iv. Iterator pattern

f) **"Encapsulate object creation"**

i. Template pattern
ii. State pattern
iii. Factory pattern
iv. Strategy pattern

g) " **Ensures only one object instance is ever created**".

i. Strategy pattern
ii. Observer pattern
iii. Singleton pattern
iv. Iterator pattern

h) Encapsulate a request as an object, thereby letting you parameterize clients with different requests, queue or log requests, and support undoable operations

i. Strategy pattern
ii. Observer pattern
iii. Command pattern
iv. State pattern

i) Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.

i. Strategy pattern
ii. Observer pattern
iii. Command pattern
iv. State pattern

j) Define a family of algorithms, encapsulate each one, and make them interchangeable.

i. Template pattern
ii. State pattern
iii. Factory pattern
iv. Strategy pattern

Q2.

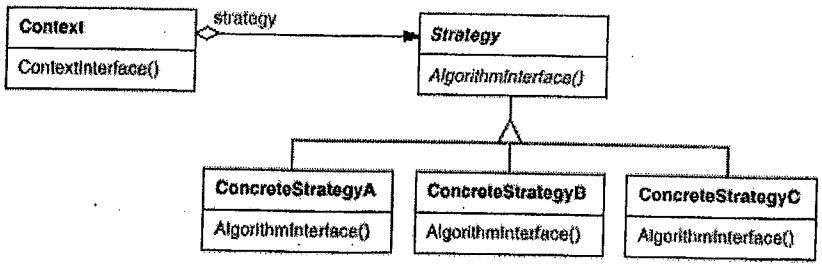
The **Struts Framework** allow low level components to hook themselves into a system but the high-level components determine when they are needed and how to use inside the framework.

(a) What is the design principal has been used in the Struts Framework?

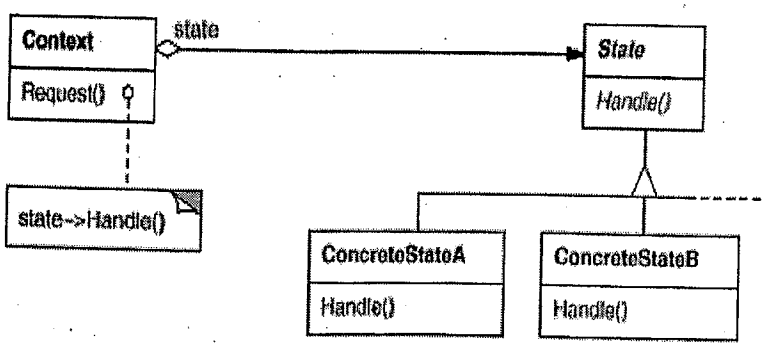
(b) Which design pattern covers above design principal?

(c) Explain how to use the above pattern in a program? You may use sample codes or class diagram to explain the pattern?

Q3.



Strategy



State

(a) What are design principals cover in above two patterns?

(b) Explain the different between state and strategy pattern you may use sample programs to describe the differences?

Q4.

```
public class Singleton {
  public static Singleton uniqueInstance;
  private static Singleton getInstance(){
       if(uniqueInstance==null){
            uniqueInstance = new Singleton();
       return uniqueInstance;
        }
    }
}
```

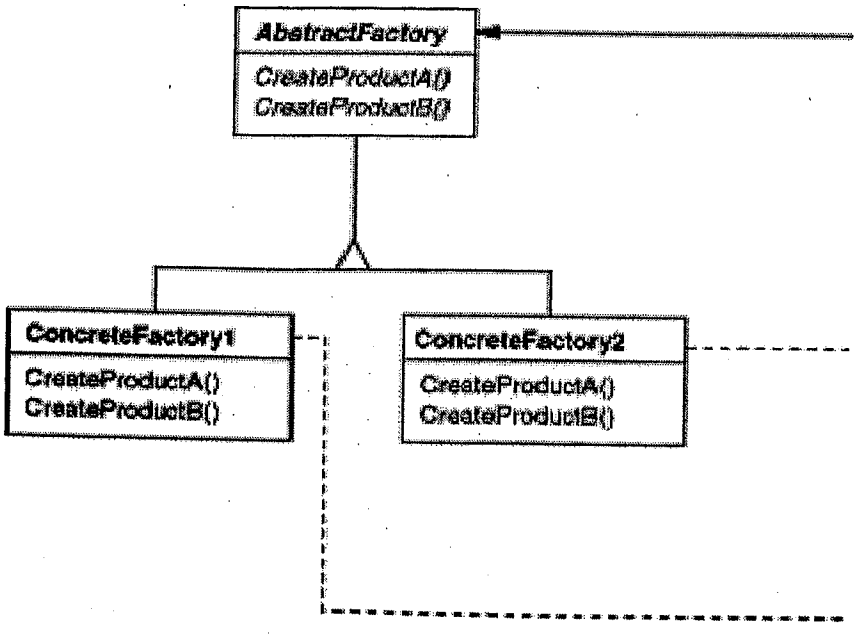
(a) What is the purpose of having singleton in a class?

(b) Above Singleton class is not working properly, Identify the errors and modify this class. How you overcome multithread issue in singleton pattern?

(c) If the application is keen on its performance, can you apply above solution to the system if yes, give the reasons, if not, modify the code and give reasons for your modifications?

Q5.

"The Abstract Factory provides an interface for a family of products "



(a) What are the advantages of defining a single interface for a family of products?
(b) Complete the above class diagram, which illustrate the abstract factory pattern?
(c) Explain Dependency Inversion Principle which covers in the factory method?