CS525 Quiz 1

Name: Student ID:

1. What is a design pattern?

2. Why are design patterns important to software designers?

3. What is a software framework?

4. Why do we need software frameworks?

5. Explain the 3 categories of GoF design patterns.

6. Name 3 design patterns for each category.

7. What is the intent of the Singleton pattern?

8. Conceptually how to implement a Singleton? Please explain with necessary code snippets.

CS525 Quiz 2

Name: Student ID:

1. What is the intent of the Adapter pattern?

2. Explain how the Adapter pattern works (you need to talk about the problem it solves, and the solution that includes structure, participants and applicability, etc.)

3. True/False: The reason we cannot change the interface of the Adaptee is because we have no access to its source code.
4. What is the intent of the Proxy pattern?

5. Explain how the Proxy pattern works (same as how the Adapter works.)

6. For the Java Applet scenario we talked about during class, how would you implement it? Just talk about ideas. (If you need more space, use the back side)

7. Provide 3 additional scenarios you know of where the Proxy pattern is used.

8. Compare the Adapter and the Proxy pattern.

CS525 Quiz 3

Name: Student ID:

1. What is the intent of the Template Method pattern?

2. You are asked to design a framework that other developers can use to quickly build a GUI application in Java. You should allow different approaches to create components on the GUI. For example, by providing code themselves, by reading from an XML file, or by loading components from a remote server, etc. But what is common are the steps to build the GUI –
   - Initialize (setRootContainer, setBackground, enableEventHandling)
   - Build components(create or load GUI components)
   - Load data (populate the data-displaying components, like textbox, dropdown, etc)
   - Apply layout (lay out the components)
   - Apply styling (css styling)
   - Display (show it on screen)

   Implement the framework with a Template method – createGUI().

3. What is the intent of the Prototype pattern?

4. What are some of the consequences of the Prototype pattern? (give at least 3)

CS525 Quiz 4

Name: Student ID:

1. What is the intent of the Factory Method pattern?

2. What are the benefits of using a Factory Method to create objects (compared with constructors)?

3. A marketing company asks you to design a document generator based on a web page (in html format). For now, they want to generate pdf files, image files, or PowerPoint files. Later there can be more formats to support by your generator. For all the current 3 document types, the company can choose to print them on A4 or A5 paper either as "portrait" or "landscape". In the future, they may also want to print them on paper of other sizes (including custom size). Design your generator and show your structure.

CS525 Quiz 5

Name: Student ID:

1. List 4 motivations of the Iterator pattern.

2. Explain how you implement an Iterator based on a given Aggregate object.

3. Explain the participants of the Composite pattern?

4. What is the main benefit of using the Composite pattern? When do you use the pattern?

5. True/False. An iterator will guarantee thread-safe concurrent traversals of aggregate objects.
6. True/False. In the Composite pattern, a Composite should know all its immediate children but its children do not know their parent.

CS525 Quiz 6 Name: Student ID:

1. When would you use the Prototype design pattern? Please choose all that apply:

   a. To abstract steps of construction of complex objects
   b. To create objects without knowing their type or any details of how to create them
   c. To avoid proliferation of the class hierarchy
   d. To restrict class instantiation to one object
   e. When creating a large/complex object that is resource intensive

2. What are the consequences of applying the Prototype pattern? Please choose all that apply:

   a. each concrete prototype class must implement the clone method
   b. it makes it easier for a certain family of objects to work together
   c. it enables the client code to register a new concrete prototype instance at run time
   d. it reduces the number of classes in the class hierarchy as compared to the factory method design pattern

3. Which of the following are participants in the Adapter design pattern? Please choose all that apply:

   a. Creator b. Product
   c. Target d. Caretaker
   e. Adapter f. Converter

4. Which design pattern can be used to implement Lazy Loading? Please choose the best one:

   a. Adapter
   b. Prototype
   c. Proxy
   d. Singleton

5. You are trying to add a class already written in another application to serve clients, beside other classes, in your system. All other classes have the same interface, the incoming class has a totally different interface from what the clients expect, but contains all required functionalities. What kind of refactoring is needed to make this class fit in with minimum changes in your system? Please choose the best one:

   a. apply the Proxy Pattern
   b. apply the Adapter Pattern
   c. create a new class which implements the expected interface and copy and paste
   d. the code from the class in the other application to this new class

6. You are building an online makeup website, which provides beside text articles, makeup tutorials as video files. Which pattern is recommended to use in such a website to deliver these media resources?

Please choose only one answer:

a. adapter
b. proxy
c. prototype
d. singleton

7. Which of the following are participants in the Composite design pattern? Please choose all that apply:

a. Aggregate
b. Component
c. Leaf
d. Iterator

8. Which of the following are participants in the Factory Method design pattern? Please choose all the answers that apply:

a. Creator
b. Product
c. Refined Abstraction
d. Abstract factory

9. When to use the Iterator Pattern? Please choose all that apply:

a. To support traversals of aggregate objects without exposing their internal representation.
b. To guarantee thread-safe, concurrent traversals of aggregate objects.
c. To provide a uniform interface for traversing different aggregate structures to support polymorphic iteration.
d. To allow different traversal methods depending on what the client needs

10. What is true about using the Template Method Pattern? Please choose all that apply:

a. It reduces code duplication.
b. It favors inheritance over composition.
c. It takes advantage of polymorphism by having the correct methods in subclasses called automatically.
d. To allow runtime change of behavior by taking different steps of operations of the template method.
e. The template method in the abstract class must be declared as 'final'.

11. Which of the following uses Chain of Responsibility to implement itself? Choose the best 2.

A java.awt.event.ComponentAdapter B The Java Servlet filter framework
C javax.jms.QueueConnectionFactory D java.awt.Toolkit E Java exception handling

12. When will you consider using the Command pattern?

    a. When you need to provide system logging for an application
    b. When you need to provide undo or system recovery capability for an application
    c. When you need to provide run-time change of behavior for an application
    d. When you want to avoid hardcoded associations between Widgets and actions to perform for a GUI application
    e. When you want to schedule execution of requests at different times

13. What is the intent of the Chain of Responsibility Pattern?

14. Draw the Structure of CoR Pattern.

15. True/False: With the CoR pattern, it is guaranteed that all requests from the client will be processed.
16. True/False: In the CoR Pattern, the chain of handlers can be dynamically changed by adding new handlers.
17. You are designing an order processing system for a web-based business. All normal domestic orders will have to be shipped by FedEx. All large bulk orders are handled separately within the warehouse and might go to different shipping carriers. Most Favored Customers (your largest customers) may have special requirements for shipping and tracking, and they're going to get what they want. All other international orders are handled in a generic manner. Draw both the class and object structures of your design and provide some skeleton code that shows how it works.

18. What is the motivation of the Command pattern?

19. Explain how Command pattern works? (remember we talked about how to explain)

20. Which methods in the Command pattern are considered "callback" methods?

CS525 Quiz 7

Name: Student ID:

1. Explain the responsibilities of each participant in the Builder pattern.

2. Template Method is to provide a skeleton algorithm for subclasses to redefine without changing the structure of the algorithm. Builder is to provide a construction process that each concrete builder follows with its own implementation of the steps. So both use a step-by-step approach to perform some functions. How do you compare the two?

3. With the Flyweight pattern, what does it mean by intrinsic and extrinsic state information?

4. Describe when the Flyweight pattern would be applicable. (5 conditions)

Quiz 2 - Microsoft Word

8. Compare the Adapter and the Proxy pattern.

Similarities:

- both use a 'wrapper' object for the real object that offers services of interest to the client;
- both provide an interface that the client expects/needs;
- both rely on object composition (or delegation) to implement it;
- both are structural patterns;

Differences:

- Their intent is different; (this is always true. If you say this, you need to explain how as below)
- Adapter is to convert an interface that a client would not otherwise be able to use.
- Proxy is to provide a surrogate or placeholder for the client to use the real subject's services.

CS525 Quiz 11

Name: Student ID:

1. What does the Context participant do in the State pattern?

2. What does the Context participant do in the Strategy pattern?

3. When will you consider using the State pattern?
   a. When an object's behavior depends on its state, and it must change its behavior at run-time depending on that state.
   b. When we want to model vending machines or implement computer video games.
   c. When you want to treat each state of an object as an object that can vary independently from others.
   d. When the number of states can change at runtime without affecting the client.
   e. When you want to localize state-specific behavior in each concrete state.
   f. When you want to break a multi-conditional statement and move related conditional branches into their own class.

4. When will you consider using the Strategy pattern?
   a. When you want the client to decouple from a strategy it uses to perform a function
   b. When you want to make the change of strategy easier at runtime.
   c. When you need to provide run-time change of behavior for an application
   d. When you want your application to use different strategies for different situations during runtime.
   e. When you want to break a multi-conditional statement and move related conditional branches into their own class.

5. Compare the State and Strategy patterns. List as many similarities and differences as you can.

CS525 Quiz 12

Name: Student ID:

1. The intent of the Bridge pattern is "Decouple an abstraction from its implementation so that the two can vary independently." What does it mean by "the two can vary independently"?

2. The Bridge pattern relies on composition to decouple abstraction from implementation. True or False?
3. We can use the Bridge pattern to make an existing implementation of an API work in our business logic. True or False?

4. What is the main difference between the Implementor and Abstraction interface?

5. When will you consider using the Decorator pattern?
   a. When you want to add responsibilities to individual objects dynamically and transparently.
   b. When you want to avoid subclassing because of a large number of independent extensions are possible and would produce an explosion of subclasses.
   c. When you need to provide run-time change of behavior for an application.
   d. When we want to change the behavior of objects of the class but not the class itself.

6. What are the advantages and disadvantages of the Decorator pattern?

CS525 Quiz 13

Name: Student ID:

1. The Java event handling model is an implementation of the Observer or Mediator pattern? Explain why.

2. "Drag and drop" is considered a UI design pattern. Can you implement it with a Mediator or Observer pattern? Explain why or why not.

3. Let's think about the Reversi board game again. We said we could implement it with a Mediator by having 2 human players modeled as 2 instances of the Colleague class. But another opinion is to model the 'positions' on the board as instances of the Colleague class. Do you agree? Please explain.

4. What are the advantages and disadvantages of the Mediator pattern?

CS525 Quiz 14

Name: Student ID:

True/False Questions 1 – 11

1. In Strategy Pattern, strategy objects are stateful.
2. Strategy Pattern helps us eliminate conditional statements from client by encapsulating the behavior in separate Strategy classes.
3. In Observer Pattern, a mediator simplifies the way colleagues communicate with each other by replacing many-to many interactions with one-to-many interactions.
4. Mediator centralizes control which can make the mediator itself a monolith that's easy to maintain.
5. Observer pattern supports layering of software applications.
6. Visitor makes adding new operations hard but adding a new Concrete Element class easy.
7. The MVC pattern decouples among the model, view, and controller.
8. In the state pattern, you usually set the state only once.
9. In the strategy pattern, the strategy is independent of the surrounding context.
10. In the strategy pattern, if a strategy does not work, there is a fallback to the next strategy (if any).
11. In the flyweight pattern, you always need both shared and unshared flyweight objects.
12. Explain Double Dispatch.

13. When will you consider using the Visitor pattern?

14. Considering the shopping cart example we saw yesterday, why it implemented the following methods, instead of visit(OrderItem orderItem);

```
public void visit(Book book);

public void visit(DVD dvd);

public void visit(Toy toy);
```

15. What is decoupled from what in each of the 8 patterns (after the midterm)?

16. Which patterns help us remove if/else statements? (all we have learned)

17. Which patterns help us avoid a lot of subclassing? (all we have learned)

18. What have you learned from the GoF patterns so far? (or what are the design 'tricks' that you have learned from all these patterns)