Design Patterns Assignment

These are the OO Principles we have looked at from HFDP

- Encapsulate what varies: Identify the aspects of your application that vary and separate them from what stays the same
- Favor composition over inheritance (Prefer a HAS-A relationship to an IS-A relationship)
- Program to interfaces, not implementations
- Strive for loosely coupled designs between objects that interact
- Open-Closed Principle: Classes should be open for extension but closed for modification
- Dependency Inversion Principle: Depend on abstractions; do not depend on concrete classes
- Single Responsibility: A class should only have one reason to change

These are the Design Patterns we have looked at from HFDP

- Strategy
- Observer
- Decorator
- Factory (Simple, Method, Abstract)
- Singleton
- Adapter
- Iterator
- Composite

Part 1

For each of the design patterns we looked at in class (see above) write the following:

- The name of the pattern
- The description of the pattern as given in the “Who Does What” exercise in chapter 13.
- The category of pattern – Creational, Structural, or Behavioral – as given in the “Sharpen Your Pencil” exercise on the next page.
- The second category of pattern – Class or Object – as given just after the first category.

Further exam study tip: The bullet points at the end of each chapter make great true/false questions!

Part 2

Answer the following questions.

1. Give the three OO design principles that the Strategy Pattern uses according to HFDP.
2. Give the three OO design principles that the Observer Pattern uses according to HFDP.
3. What are observers called in the Swing framework? (see “Other places you’ll find the Observer Pattern in the JDK” in Chapter 2)
4. What is the main OO design principle introduced in the discussion of the Decorator Pattern?
5. The Java I/O package is largely based on decorator. InputStream represents the abstract component. What class represents the corresponding abstract decorator? (see Decorating the java.io classes in chapter 3)
6. What is the main OO design principle introduced in the discussion of the Factory Method Pattern?
7. What OO design principle does the Factory Method Pattern ignore? (see bullet points in chapter 4)
8. What kind of class (introduced in Java 1.5) does “Effective Java” by Joshua Block recommend for creating singleton objects in Java?
9. What keyword (instead of class) does Scala to create a singleton object?
10. There are two forms of the Adapter Pattern: object and class adapters. Which one did we cover? (see bullet points for chapter 7)
11. What is the main OO design principle introduced in the discussion of the Iterator Pattern?
12. What OO design principle does the Composite Pattern violate? (see the section just before “Flashback to Iterator” in chapter 9)
13. The Java Swing components rely heavily on the Composite Pattern. Consider the following four JComponents: JPanel, JButton, JLabel, JMenu. Which are typically used as leaf classes and which are typically used as composite classes?
14. Look at the Builder Pattern in HFDP. Give its brief description; tell whether it is a creational, structural, or behavioral pattern; and tell whether it is a class or object pattern.
15. What is the name of the Builder for the String class? What is the name of its primary method? Is the builder class mutable or immutable?