

Name _____

Solve any 8 of the following 11 problems. Indicate which 8 problems you are solving. Only 8 problems will be graded.

1. We have a Student class which has a method grade(). This method returns an integer value, 4 for an A, 3 for a B, 2 for a C, 1 for a D and 0 for an F. We are going to write code that will select students from a list with grades in a given range, for example grades between 1 and 3. What conditions would you check for in your unit tests?

2 Select one of the following patterns – Strategy, Composite or Iterator and discuss the consequences (good and bad) of the pattern.

3 The design patterns text uses two main design principles. Select one pattern that uses both principles. Show how it uses both.

4. A singleton is considered more flexible than implementing all operations as class operations (static methods in C++ & Java, class methods in Smalltalk). How is the singleton more flexible?
- 5a. In the XML parser what roles in the Builder pattern did the classes XMLParser, ContentHandler, and CS635Document play?
- 5b. The Builder pattern is said to reduce algorithmic dependencies. Explain how the builder reduced algorithmic dependencies in the XML parser assignment.

6. One issue in the composite pattern is who should declare the child management operations: the Composite or the Component classes. What are the trade-offs involved in deciding where to declare the child management operations.

7. Show how the Null object pattern can be used to replace if (or case) statement.

8. The Object Adapter and Proxy are similar in structure: both wrap an object. What is the difference between the Adapter and Proxy patterns?

9a. What is an external iterator?

9b. What is an internal iterator?

9c. Which is easier to implement on a Composite structure like a binary search tree?

10. Sometimes it can be difficult to tell if one should use the State or the Strategy pattern. Give some guidelines to tell when to use the State and when to use the Strategy pattern.

11. The Visitor pattern has an interesting way to determine the correct visit method to use in a visitor object. Explain how in the Visitor pattern the correct visit method in the correct Visitor class is called when a visitor traverses a structure.