

2015

Answer all 14 questions. Answer essay questions as briefly as possible.

The following might be names of patterns: Abstract Class, Abstract Factory, Active Object Model, Adapter, Application Controller, Bridge, Builder, Chain of Responsibility, Collaborator, Command, Composite, Decorator, Dependency Injection, Dynamic Factory, Extension Object, Façade, Factory Method, Flyweight, Interpreter, Iterator, Master-Slave, Mediator, Memento, MVC, Null Object, Observer, Property, Prototype, Proxy, Singleton, Schema, Smart Variables, Specification, State, Strategy, Template Method, Type Object, Value Object, Visitor.

1. (2 points) What design pattern might you use when you wish to reduce tight coupling between classes?
2. (2 points) What design pattern would you use to change the algorithm that an object uses?
3. (2 points) What design pattern would you use to change the kind and number of objects that react to changes in the object you are designing?
4. (2 points) What design pattern would you use to add operations to classes without changing the class?
5. (2 points) What design pattern would you use to change how methods in a class behave?
6. (8 points) What is a code smell? List two code smells.
7. (12 points) Circle the correct answer for each of the following.

True	False	In the interpreter pattern is it easy to extend grammar.
True	False	The interpreter pattern is not efficient.
True	False	The Null Object pattern is not used with the Iterator pattern.
True	False	The Singleton is more flexible than class operations.
True	False	In the State pattern the Context should define the state transitions.
True	False	State objects can be shared with different Context objects.
True	False	In the Visitor pattern adding new ConcreteElement classes is easy.

- |      |       |  |
|------|-------|--|
| True | False | Command objects can perform the operation itself without delegating to a receiver. |
| True | False | Internal iterators are easier to implement than external iterators.                |
| True | False | The singleton pattern permits a variable number of instances.                      |
| True | False | Mementos might be expensive.   |
| True | False | The Memento pattern allows the Originator to store a history of mementos.          |

8. (10 points) Explain the justification of the Shearing Layers pattern in the Big Ball of Mud paper.
9. (10 points) Explain one of the following types of coupling: Data Coupling, Control Coupling, Inside Internal Object Coupling.
10. (10 points) The design patterns text claims it uses two main design principles. Select one pattern that uses both principles. Show how it uses both.
11. (10 points) Select either the Visitor pattern or the Decorator pattern and explain how the pattern works.
12. (10 points) Explain one of the following types of cohesion: Logical, Temporal, Procedural, Communication, Sequential. Give an example.
13. (10 points) 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.
14. (10 points) Under which conditions does it make sense to use the Null Object pattern?

2014

The following might be names of patterns: Abstract Class, Abstract Factory, Adapter, Application Controller, Bridge, Builder, Chain of Responsibility, Collaborator, Command, Composite, Decorator, Façade, Factory Method, Flyweight, Interpreter, Iterator, Master-Slave, Mediator, Memento, Null Object, Observer, Prototype, Proxy, Singleton, Specification, State, Strategy, Template Method, Visitor.

15. The purpose of many of the design patterns is to make it easy to change some property of the system. Give one design pattern you would use to make it easy to change the following.
  - A. The algorithm that an object uses.
  - B. The class of the object that a method returns.
  - C. The kind and number of objects that react to changes in the object you are designing.
  - D. Adding operations to classes without changing the class.
  - E. How methods in a class behave.
16. What is a code smell? List two code smells.
17. Under which conditions does it make sense to use the Null Object pattern?
18. The design patterns text claims that two main design principles are used in design patterns. Give one of the design principles. What are the benefits and drawbacks of using this principle.
19. Explain one of the following types of cohesion: Logical, Temporal, Procedural, Communication, Sequential. Give an example.
20. Explain how the Observer pattern works.
21. What is the "Piecemeal Growth" pattern? What are the advantages and disadvantages of the pattern?
22. How are the Proxy & Command patterns similar? How are they different?
23. a. Explain object-oriented recursion.
  - b. List one pattern that can be considered a specialization of object-oriented recursion.
24. Explain one of the following types of coupling: Data Coupling, Control Coupling, Inside Internal Object Coupling.
25. (12 points) Circle the correct answer for each of the following.

2013

- |      |       |  |
|------|-------|--|
| True | False | In the interpreter pattern is it easy to extend grammar.                           |
| True | False | The interpreter pattern is not efficient.  |
| True | False | The Null Object pattern is not used with the Iterator pattern.                     |
| True | False | The Singleton is more flexible than class operations.                              |
| True | False | In the State pattern the Context should define the state transitions.              |
| True | False | State objects can be shared with different Context objects.                        |
| True | False | In the Visitor pattern adding new ConcreteElement classes is easy.                 |
| True | False | Command objects can perform the operation itself without delegating to a receiver. |
| True | False | Internal iterators are easier to implement than external iterators.                |
| True | False | The singleton pattern permits a variable number of instances.                      |
| True | False | Mementos might be expensive.   |
| True | False | The Memento pattern allows the Originator to store a history of mementos.          |

26. What is a code smell? List two code smells.

The following might be names of patterns: Abstract Class, Abstract Factory, Active Object Model, Adapter, Application Controller, Bridge, Builder, Chain of Responsibility, Collaborator, Command, Composite, Decorator, Dependency Injection, Dynamic Factory, Extension Object, Façade, Factory Method, Flyweight, Interpreter, Iterator, Master-Slave, Mediator, Memento, MVC, Null Object, Observer, Property, Prototype, Proxy, Singleton, Schema, Smart Variables, Specification, State, Strategy, Template Method, Type Object, Value Object, Visitor.

27. (2 points) What design pattern might you use when you wish to reduce tight coupling between classes?

28. (2 points) What design pattern would you use to change the algorithm that an object uses.

29. (2 points) What design pattern would you use to change the kind and number of objects that react to changes in the object you are designing.

30. (2 points) What design pattern would you use to add operations to classes without changing the class.
31. (10 points) Under which conditions does it make sense to use the Null Object pattern?
32. (10 points) In object-oriented programming one is advised to avoid case (and if) statements. Select one design pattern that helps avoid case statements and explain how it helps.
33. (10 points) Both the Memento and Command patterns can be used to implement undo. What are the disadvantages and advantages of each pattern for implementing undo.
34. (10 points) Explain one of the following types of coupling: Data Coupling, Control Coupling, Inside Internal Object Coupling.
35. (10 points) Explain one of the following types of cohesion: Logical, Temporal, Procedural, Communication, Sequential. Give an example.
36. (10 points) 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.
37. (10 points) What is the "Piecemeal Growth" pattern? What are the advantages and disadvantages of the pattern?

2012

The following might be names of patterns: Abstract Class, Abstract Factory, Adapter, Application Controller, Bridge, Builder, Chain of Responsibility, Collaborator, Command, Composite, Decorator, Façade, Factory Method, Flyweight, Interpreter, Iterator, Master-Slave, Mediator, Memento, Null Object, Observer, Prototype, Proxy, Singleton, Specification, State, Strategy, Template Method, Visitor.

38. The purpose of many of the design patterns is to make it easy to change some property of the system. What design pattern would you use to make it easy to change the following.
- A. The algorithm that an object uses.
  - B. The class of the object that a method returns.
  - C. The kind and number of objects that react to changes in the object you are designing.
  - D. Adding operations to classes without changing the class.
  - E. How methods in a class behave.
39. The design patterns text claims that two main design principles are used in design patterns. Give one of the design principles. What are the benefits and drawbacks of using this principle.
40. Under which conditions does it make sense to use the Null Object pattern?
41. Explain.
- A. Information Hiding
  - B. Encapsulation
  - C. Abstraction
42. Explain one of the following types of cohesion: Logical, Temporal, Procedural, Communication, Sequential. Give an example.
43. Explain how the Observer pattern works.
44. What is the "Piecemeal Growth" pattern? What are the advantages and disadvantages of the pattern?
45. How are the Proxy & Command patterns similar? How are they different?
46. Explain object-oriented recursion.
- b.** List one pattern that can be considered a specialization of object-oriented recursion.
47. Explain one of the following types of coupling: Data Coupling, Control Coupling, Inside Internal Object Coupling.

48. (10 points) Circle the correct answer for each of the following.

- |      |       |   |
|------|-------|---|
| True | False | In the builder pattern the director selects the concrete builder.                                 |
| True | False | The builder pattern lets you vary a product's internal representation.                            |
| True | False | In the chain of responsibility more than one handler can handle a particular request.             |
| True | False | In the chain of responsibility a request will be handled.   |
| True | False | Chain of responsibility should be use when who handles a request should be specified dynamically. |
| True | False | The interpreter pattern can be used for complex grammars.   |
| True | False | In the interpreter pattern is it easy to extend grammar.  |
| True | False | The interpreter pattern is not efficient.   |
| True | False | A prototype manager is required in the prototype pattern.   |
| True | False | The prototype pattern can reduce subclassing.   |

49. (10 points) Circle the correct answer for each of the following.

- |      |       |  |
|------|-------|--|
| True | False | In the proxy pattern the proxy does not create the real subject.                       |
| True | False | The singleton pattern is an improvement over global variables.                         |
| True | False | The singleton pattern permits a variable number of instances.                          |
| True | False | The abstract factory pattern promotes consistency among products.                      |
| True | False | Supporting new kinds of products in difficult when using the abstract factory pattern. |
| True | False | In implementing an abstract factory the singleton pattern is often used.               |
| True | False | Mementos might be expensive.   |
| True | False | The Memento pattern allows the Originator to store a history of mementos.              |
| True | False | The composite pattern can make your design overly general.                             |
| True | False | The composite pattern makes the client simple.   |

The following might be names of patterns: Abstract Class, Abstract Factory, Active Object Model, Adapter, Application Controller, Bridge, Builder, Chain of Responsibility, Collaborator, Command, Composite, Decorator, Dependency Injection, Dynamic Factory, Façade, Factory Method, Flyweight, Interpreter, Iterator, Master-Slave, Mediator, Memento, Null Object, Observer, Property, Prototype, Proxy, Singleton, Schema, Smart Variables, Specification, State, Strategy, Template Method, Type Object, Visitor.

50. (2 points) What design pattern would you use to make it easy to change the implementation of an abstraction?
51. (2 points) What design pattern should you think of when you want to hide how you construct a complex object?
52. (2 points) What design pattern should you think of when when you wish to send a method to one of several objects without specifying the receiver explicitly?
53. (2 points) What design pattern would you use when you have a group of related objects that are designed to work together and you need to insure that they are used together?
54. (2 points) Which design pattern would you use when you want a client to create a new object without explicitly specifying the class of the new object?
55. (2 points) What design pattern would you use to make it easy to change the algorithm a class uses?
56. (2 points) What design pattern would you use when more that one object may handle a request and you don't know in advance which object will handle a particular request?
57. (2 points) What design pattern might you use when you wish to reduce tight coupling between classes?
58. (2 points) What design pattern would you use when you only have the binary of a class and need to modify the signature of some of the methods in the class?
59. (2 points) What design pattern would you use to reduce dependence on hardware and/or software platforms?



60. (10 points) The text claims that Mediator and Observer are competing patterns.
- Explain how they compete with each other.
  - What are the strengths of each pattern over the other.
61. (10 points)
- What is the difference between intrinsic and extrinsic state?
  - Give an example of each.
62. (10 points) Design patterns have consequences, some good and some bad.
- Give one good consequence of the Singleton pattern.
  - Give one bad (or negative) consequence of the Singleton pattern.
63. (10 points) In object-oriented programming one is advised to avoid case (and if) statements. Select one design pattern that helps avoid case statements and explain how it helps.
64. (10 points) Decorator and Chain of Responsibility are similar because they both contain multiple nested classes. In other respects, they are quite different. Explain the differences between Decorator and Chain of Responsibility.

## 2010 Midterm Exam

Answer all 10 questions. Answer essay questions as briefly as possible.

The following might be names of patterns: Abstract Class, Abstract Factory, Adapter, Application Controller, Bridge, Builder, Chain of Responsibility, Collaborator, Command, Composite, Decorator, Façade, Factory Method, Flyweight, Interpreter, Iterator, Master-Slave, Mediator, Memento, Null Object, Observer, Prototype, Proxy, Singleton, Specification, State, Strategy, Template Method, Visitor.

65. The purpose of many of the design patterns is to make it easy to change some property of the system. What design pattern would you use to make it easy to change:

- A. The algorithm that an object uses.
- B. The class of the object that a method returns.
- C. The kind and number of objects that react to changes in the object you are designing.
- D. Adding operations to classes without changing the class.
- E. How methods in a class behave.

66. What is a code smell? List two code smells.

67. Explain shallow copy and deep copy.

68. Explain control coupling. Give an example.

69. Explain Coupling and Cohesion.

70. According to the Big Ball of Mud paper why should we use piecemeal growth?

71. Explain the command processor pattern.

72. What is the difference between the Proxy and Decorator pattern?

73. Explain how the State pattern works.

74. What are some of the problems caused by using the singleton pattern?