

ECE 417/617 Assignment #6

Read Chapter 6 of Steve McConnell, *Code Complete*, 2nd edition, Microsoft Press, 2004.

Then answer the following questions.

Chapter 6: Classes

43. Over the years, thoughts about programming have evolved from _____ to _____ to _____.
44. What is exciting about Abstract Data Types (ADTs)?
45. Using ADTs allows you to hide _____.
46. What mistake is made by most discussions of ADTs?
47. What is wrong with the interface on p. 134? On the bottom of p. 135?
48. An aspect of an interface that cannot be enforced by the compiler should be converted from a _____ interface element to a _____ interface element.
49. Which is more correct?
 - (a) If a class has good abstraction, it will likely have strong cohesion.
 - (b) If a class has strong cohesion, it will likely have good abstraction.Therefore, should a programmer focus more upon abstraction or cohesion?
50. C++ has the drawback of requiring programmers to disclose implementation details in the class interface. What technique does Scott Meyers suggest to overcome this limitation? (Note: Elsewhere this is known as an *opaque pointer*.)
51. What is the danger of looking at a class's implementation?
52. What are the two primary ways of having two classes relate to each other (hint: one enforces "has a", the other "is a")?
53. How many discrete items can a person remember while performing other tasks?
54. Suppose three classes derive from a base class, and any of the routines in the base class can be called without worrying about which derived class is being used. What principle does this code follow?

55. Of the three flavors of inherited routines, I have found the abstract overridable routine to be the most useful. Can you think of any drawbacks of the other two?
56. What happens if a derived class has a function with the same name as a non-overridable routine in the base class?
57. A programmer creates a base class with a single derived class. What is wrong with this approach of anticipating future needs?
58. What is the maximum recommended depth of an inheritance tree?
59. Multiple inheritance is useful with “mixins”, where one base class is a noun and one is an adjective. Give some example adjectives that one might encounter in this context.
60. As a programmer, you should maintain a heavy bias against _____.
61. What is the “Law of Demeter”?
62. In a constructor, should you get into the habit of initializing all the data members?
63. What is McConnell’s opinion of reference counting?
64. What did NASA discover when it conducted a study to compare the code reuse of object-oriented versus functionally-oriented projects?