Week 5 Review – Abstraction and Polymorphism

1. All abstract classes must contain abstract methods.

a. True b. False

2. Non-abstract classes are called \_\_\_\_\_\_\_ classes.

a. real b. instance c. implementable d. concrete

3. The following object declaration for the abstract class GraphicalObject is a legal statement.

GraphicalObject g;

g = new GraphicalObject (new Point (0,0));

a. True b. False

4. Which statements are true about abstract superclasses is *true*?

a. abstract superclasses may contain data.

b. abstract superclasses may *not* contain implementations of methods.

c. abstract superclasses must declare all methods as abstract.

d. abstract superclasses must declare *all* data members not given values as

abstract.

5. Consider the abstract superclass below:

public abstract class Foo {  
 private int a;  
 public int b;  
  
 public Foo( int aVal, int bVal ) {  
 a = aVal;  
 b = bVal;  
 } // end Foo constructor  
  
 public abstract int calculate();  
} // end class Foo

Any *concrete* subclass that *extends* class Foo:

a. Must implement method calculate b. Will *not* access instance variable a

c. Neither (a) nor (b) d. Both (a) and (b)

6. Consider classes A, B and C, where A is an abstract superclass, B is a concrete class that inherits from A and C is a concrete class that inherits from B. Class A declares abstract method originalMethod, implemented in class B. Which of the following statements is *true* of class C?

a. Method originalMethod cannot be overridden in class C—once it has been implemented in concrete class B, it is implicitly final.

b. Method originalMethod *must be* overridden in class C, or a syntax error will occur.

c. If method originalMethod is not overridden in class C but is called by an object of class C, an error occurs.

d. None of the above.

7. When a superclass variable refers to a subclass object and a method is called on that object, the proper implementation is determined at execution time. What is the process of determining the correct method to call?

a. early binding b. non-binding c. on-time binding d. late binding

8. Polymorphism enables you to:

a. program in the general.

b. program in the specific.

c. absorb attributes & behavior from previous classes.

d. hide details from user.

9. For which of the following would polymorphism *not* provide a clean solution?

a. A billing program where there is a variety of client types that are billed with different fee structures.

b. A maintenance log program where data for a variety of types of machines is collected and maintenance schedules are produced for each machine based on the data collected.

c. A program to compute a 5% savings account interest for a variety of clients.

d. An IRS program that maintains information on a variety of taxpayers and determines who to audit based on criteria for classes of taxpayers.

10. Polymorphism allows for specifics to be dealt with during:

a. program execution b. compilation c. programming d. debugging