Week 3b Review **OOP Concepts**: **Abstraction** (Classes and Methods)

Through abstraction, a programmer *hides* all but the relevant data about an [object](http://searchsoa.techtarget.com/definition/object) in order to reduce complexity (i.e., only providing a set of essential characteristics) and thus increase efficiency.

1. A(n) class cannot be instantiated.

a. final.

b. concrete.

c. abstract.

d. polymorphic.

2. Non-abstract classes are called:

a. real classes.

b. instance classes.

c. implementable classes.

d. concrete classes.

3.Which of the following could be used to declare abstract method method1 in abstract class Class1 (method1 returns an int and takes no arguments)?

a. public int method1();

b. public int abstract method1();

c. public abstract int method1();

d. public int nonfinal method1();

4. Which of the following statements 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 a method called calculate.

b. Will not be able to access the instance variable a.

c. Neither (a) nor (b).

d. Both (a) and (b).