SELF TEST

The following questions will help you measure your understanding of the material presented in this chapter. Don't even *think* about skipping this test. You really need to see what the questions on the exam can be like, and check your grasp and memorization of this chapter's topics.

Encapsulation, IS-A, HAS-A (Sun Objective 6.1)

1. Given the following,

```
1.    public class Barbell  {
2.       public int getWeight ( )  {
3.             return weght;
4.       }
5.       public void setWeight (int w)  {
6.             weight = w;
7.       }
8.       public int weight;
9.    }
```

Which is true about the class described abobe?

A. Class Barbell is tightly encapsulated.
B. Line 2 is in conflict with encapsulation.
C. Line 5 is in conflict with encapsulation.
D. Line 8 is in conflict with encapsulation.
E. Lines 5 and 8 are in conflict with encapsulation.
F. Lines 2, 5, and 8 are in conflict with encapsulation.

2. Given the following,

```
1.    public class B extends A  {
2.       private int bar;
3.       public void setBar (int b)  {
4.             bar = b;
5.       }
6.    }
7.    class A  {
8.       public int foo;
9.    }
```

Which is true about the classes described above?

A. Class A is tightly encapsulated.
B. Class B is tightly encapsulated.
C. Classes A and B are both tightly encapsulated.
D. Neither class A nor class B is tightly encapsulate.

3. Which is true?

A. Tightly encapsulated classes are typically easier to reuse.
B. Tightly encapsulated classes typically use inheritance more than unencapsulated classes.
C. Methods in tightly encapsulated classes cannot be overridden.
D. Methods in tightly encapsulated classes cannot be overloaded.
E. Tightly encapsulated classes typically do not use HAS-A relationships.

4. Which two are *not* benefits of encapsulation? (Choose two.)

A. Clarity of code
B. Code efficiency
C. The ability to add functionality later on
D. Modifications require fewer coding changes
E. Access modifiers become optional

5. Given the following,

```
class B extends A  {

   int getID ( )  {
         return id;
   }
}
class C  {
   public int name;
}
class A  {
   C c = new C ( );
   public int id;
}
```

Which two are true about instances of the classes listed above? (Choose two.)

A. A is-a B
B. C is-a A
C. A has-a C
D. B has-a A
E. B has-a C

Overriding and Overloading (Sun Objective 6.2)

6. Given the following,

```
Class A  {
  public void baz ( )  {
        System.out.println (“A”);
  }
}
public class B extends A  {
   public static void main (String [ ] args)  {
         A a = new B ( );
         a.baz ( );
   }
   public void baz ( )  {
         System.out.println (“B”);
   }
}
```

What is the result?

A. A
B. B
C. Compitation fails.
D. An exception is thrown at runtime.

7. Given the following,

```
class Foo  {
   String doStuff (int x)  {  return “hello”;  }
}
```

Which method would not be legal in a subclass of Foo?

A. String doStuff (int x) { return “hello”; }
B. int doStuff (int x) { return 42; }
C. public String doStuff (int x) { return “Hello”; }
D. protected String doStuff (int x) { return “Hello”, }
E. String doSruff (String s) { return “Hello”; }
F. int doStuff (String s) { return 42; }

8. Given the following,

```
1.        class ParentClass  {
2.           public int doStuff (int x)  {
3.                 return x * 2;
4.           }
5.        }
6.
7.        public class ChildClass extends ParentClass  {
8.           public static void main (String [ ] args )  {
9.                 ChildClass cc = new ChildClass ( ) ;
10.                long x = cc.doStuff (7);
11.                System.out.println (“x = “ + x);
12.          }
13.
14.          public long doStuff (int x)  {
15.                return x * 3;
16.        }
17.     }
```

What is the result?

A. x = 14
B. x = 21
C. Compilation fails at line 2
D. Compilation fails at line 11
E. Compilation fails at line 14
F. An exception is thrown at runtime.

9. Given the following,

```
1.      class Over  {
2.         int doStuff (int a, float b)  {
3.               return 7;
4.         }
5.      }
6.
7.      class Over2 extends Over   {
8.               // insert code here
9.      }
```

Which two methods, if inserted independently at line 8, will not compile? (Choose two.)

A. public int doStuff (int x, float y) { return 4; }
B. protected int doStuff (intx, float y) { return 4; }
C. private int doStuff (int x, float y) { return 4; }
D. private int doStuff (int x, double y) { return 4; }
E. long doStuff (int x, float y) { return 4; }
F. int doStuff (float x, int y) { return 4; }

10. Given the following,

```
1.      public class TestPoly  {
2.         public static void main (String [ ] args )  {
3.               Parent p = new Child ( );
4.         }
5.      }
6.
7.      class Parent  {
8.         public Parent ( )  {
9.               super ( );
10.              System.out.println (“instantiate a parent”);
11.        }
12.     }
13.
14.     class Child extends Parent  {
    public Child ( )  {
        System.out.println (“instantiate a child”);
    }
}
```

What is the result?

A. instantiate a child
B. instantiate a parent
C. instantiate a child
instantiate a parent
D. instantiate a parent
instantiate a child
E. Compilation fails
F. An exception is thrown at runtime

11. Given the following,

```
public class TestPoly  {
    public static void main (String [ ] args )  {
        Parent p = new Child ( );
    }
}

class Parent  {
    public Parent ( )  {
        super ( );
        System.out.println (“instantiate a parent”);
    }
}

class Child extends Parent  {
    public Child ( )  {
        System.out.println (“instantiate a child”);
        super ( );
    }
}
```

What is the result?

A. instantiate a child
B. instantiate a parent
C. instantiate a child
instantiate a parent
D. instantiate a parent
instantiate a child
E. Compilation fails.
F. An exception is thrown at runtime.

12. Given the following,

```
1.     class MySuper  {
2.        public MySuper (int i)  {
3.              System.out.println (“super “ + i);
4.        }
5.     }
6.
7.     public class MySub extends MySuper  {
8.        public MySub ( )  {
9.              super (2) ;
10.             System.out.println (“sub”)  {
11.       }
12.
13.       public static void main (String [ ] args)  {
14.             MySuper sup = new MySub ( );
15.       }
16     }
```

What is the result?

A. sub
super 2
B. super 2
sub
C. Compilation fails at line 2.
D. Compilation fails at line 8
E. Compilation fails at line 9
F. Compilation fails at line 14

13. Given the following,

```
1.     public class ThreeConst  {
2.        public static void main (String [ ] args)  {
3.              new ThreeConst (4L);
4.        }
5.        public ThreeConst (intX)  {
6.              this ( );
7.              System.out.print (“ “ + (x * 2) );
8.        }
9.        public ThreeConst (long x)  {
10.             this ( (int) x) ;
11.             System.out.print (“ “ + x);
12.       }
13.
14.       public ThreeConst ( )  {
15.             System.out.print (“no-arg “);
16.       }
17.    }
```

What is the result?

A. 4
B. 4 8
C. 8 4
D. 8 4 no-arg
E. No-arg 8 4
F. Compilation fails.

14. Given the following,

```
public class ThreeConst  {
    public static void main (String [ ] args)  {
            new ThreeConst ( );
    }
    public void ThreeConst (int x)  {
            System.out.print (" " + (x * 2));
    }
    public void ThreeConst (long x)  {
       System.out.print (" " + x)  {
    }

    public void ThreeConst ( )  {
            System.out.print ("no-arg ");
    }
}
```

What is the result?

A. no-arg
B. 8 4 no-arg
C. no-arg 8 4
D. Compilation fails.
E. No output is produced.
F. An exception is thrown at runtime.

15. Given the following,

```
class Dog  {
    Dog (String name)  { }
}
```

If class Beagle extends Dog, and class Beagle has only one constructor, which of the following could be the legal constructor for class Beagle?

A. Beagle ( ) { }
B. Beagle ( ) { super ( ) ; }
C. Beagle ( ) { super (“fido”) ; }
D. No constructor, allow the default constructor

16. Which two of these statements are true about constructors? (Choose two.)

A. Constructors must not have arguments if the superclass constructor does not have arguments.
B. Constructors are not inherited.
C. Constructors cannot be overloaded.
D. The first statement of every constructor is a legal call to the super ( ) or this ( ) method.

Return Types (Sun Objective 1.4)

17. Given the following,

```
13.     int x;
14.     x = n.test ( );
18.     int test ( )  {
19.
20.         return y;
21.     }
```

Which line of code, inserted at line 19, will not compile?

A. short y = 7
B. int y = (int) 7.2d;
C. Byte y = 7;
D. char y = ‘s’;
E. int y = 0xface;

18. Given the following,

```
14.     long test ( int x, float y)  {
15.
16.     }
```

Which two of the following lines, inserted independently, at line 15 would not compile? (Choose two.)

A. return x;
B. return (long) x / y;
C. return (long) y;
D. return (int) 3.14d;
E. return (y / x) ;
F. return x / 7;

19. Given the following,

```
1.      import java.util.*;
2.      class Ro  {
3.          public static void main (String [ ] args)  {
```

```
4.                Ro r = new Ro ( );
5.                Object o = r.test ( );
6.            }
7.
8.            Object test ( )  {
9.
10.
11.           }
12.       }
```

Which two of the following code fragments inserted at lines 9/10 will not compile? (Choose two.)

A. return null;
B. Object t = new Object ( );
return t;
C. int [ ] a = new int [2];
return a;
D. char [ ] [ ] c = new char [2] [2];
return c[0] [1];
E. char [ ]   [ ] c = new char [2] [2];
return c[1];
F. return 7;

20. Given the following,

```
1.        import java.util.*;
2.        class Ro  {
3.            public static void main (String [ ] args)  {
4.                Ro r = new Ro ( );
5.                Object o = r.test ( );
6.            }
7.
8.            Object test ( )  {
9.
10.
11.           }
12.       }
```

Which two of the following code fragments inserted at lines 9/10 will not compile? (Choose two.)

A. char [ ] [ ] c = new char [2] [2];
return c;
B. return (Object) 7;
C. return (Object) (new int [ ] {1, 2, 3});
D. ArrayList a = new ArrayList ( );
return a;
E. return (Object) “test”;
F. return (Float) 4.3;

21. Given the following,

```
class Test  {
    public static Foo f = new Foo ( );
    public static Foo f2;
    public static Bar b = new Bar ( );

    public static void main (String [ ] args)   {
            for (int x=0; x<6; x++)  {
                f2 = getFoo (x);
                f2.react ( );
        }
    }
    static Foo getFoo(int y)  {
              if ( 0 = = y % 2 )  {
                        return f;
              } else  {
                        return b;
              }
    }
}

class Bar extends Foo   {
    void react ( )  { System.out.print (“Bar “);  }
}

class Foo  {
    void react ( )  { Syste.out.print (“Foo “);  }
}
```

What is the result?

A. Bar Bar Bar Bar Bar Bar
B. Foo Bar Foo Bar Foo Bar
C. Foo Foo Foo Foo Foo Foo
D. Compilation fails.
E. An exception is thrown at runtime.