1. Given the following variable expression

Object obj = new String(“hello”);

1. obj.upperCase();
2. obj.subString(0,5);
3. obj.equals(**"hi"**);
4. obj.compareTo(**"hi"**);
5. All of them will cause compilation errors.

Multiple choices possible.

True/False:

1. A polymorphic reference variable can refer to different types of objects at different times.
2. Polymorphism may occur with inheritance but not with interfaces.

Multiple Choice:

**public class** Abc {  
  
 **public int a**;  
 **public static final int *b***=0;  
  
 **public void** c(){  
 System.***out***.println(**"ABC"**);  
 }  
}

**public class** Xyz **extends** Abc {  
  
 **public void** c(){  
 **super**.c();  
 System.***out***.println(**"XYZ"**);  
 }  
  
 **public void** z(){  
 c();  
 }  
}

1. Which statement best characterizes the relationship between the two classes ?
   1. An Abc is an Xyz
   2. An Xyz is an Abc
   3. An Abc has an Xyz
   4. An Xyz has an Abc
   5. Abc and Xyz are siblings
2. Which members of the class Abc can be used directly by the class Xyz ?
   1. a
   2. b
   3. c
3. I only
4. II only
5. I and III only
6. II and III only
7. I,II and III only.
8. What is printed by the following statements?

Xyz q = new Xyz();

q.c();

1. ABC
2. XYZ
3. ABCXYZ
4. XYZABC
5. Nothing is printed because there is a compile error.
6. Consider the following statement.

ArrayList<Abc> items = **new** ArrayList<Abc>();

Which of the following statements will produce a compile error?

1. items.add(**new** Abc());
2. items.add(new Xyz());
3. items.add(new Object());
4. None will produce a compile error.
5. II only
6. III only
7. I and II only
8. II and III only
9. Consider the following statements.

Abc p = **new** Xyz();  
System.***out***.println(*//missing);*

Which of the following , if used to replace missing above will cause an error?

* 1. p.b
  2. p.c
  3. p.z
  4. p
  5. p.toString()

Multiple answers are possible.

**public interface** Speaker {  
  
 **public void** speak();  
}

**public interface** Writer {  
  
 **public void** write();  
}

**public class** Philosopher **implements** Speaker,Writer {  
 @Override  
 **public void** write() {  
  
 }  
  
 @Override  
 **public void** speak() {  
  
 }  
  
 **public static void** main(String[] args) {  
  
 Object o = **new** Writer();  
 }  
}

* 1. Speaker s = **new** Philosopher();
  2. Speaker s1 = **new** Writer();
  3. Philosopher p = **new** Speaker();;
  4. Philosopher p1 = **new** Writer();

Which of the above will cause an error?