1. In order to be written to a stream using ObjectOutputStream, a class:
   a. Cannot have any fields declared as transient
   b. Cannot contain any static fields or methods
   c. Must provide a SerialPresistentFields array.
   d. Must have all fields and methods declared public
   e. Must implement the Serializable interface.

2.
```
public class SaveObject {
   public static void main(String args[]) {
         Employee smith = new
                     Employee(“012345”,”Smith”,”James”,”Payroll”);
         FileOutputStream fos = new FileOutputStream(“smith.dat”);
         ObjectOutputStream oos = new ObjectOutputStream(fos);
         oos.writeObject(smith);
         fos.close();
    }
}
```

   What action do you take for the sample code above to compile?

   a. Implement the class Employee as Serializable
   b. Do not declare the class Employee as transient
   c. Do not implement the class Employee as Externalizable.
   d. Do not create the file “smith.dat”.
   e. Create the file “smith.dat”.

3. Sample Code

```
Line 1 FileInputStream f = new FileInputStream("store");
Line 2 ObjectInputStream in = new ObjectInputStream(f);
Line 3 Object obj = in.readObject();
Line 4 Additional code here
```

Based on the sample code above, which code do you insert in place of "Additional code here" to discover the type of object represented by obj?

a. obj.getClass();
b. new obj.instanceOf();
c. ClassLoader.getInstance(obj);
d. new Class.getName(obj);
e. obj.equals();

4. Sample Code

```
1. MyObject myObject = new MyObject();
2. FileOutputStream fos = new FileOutputStream("myobject.ser");
3. ObjectOutputStream oos = new ObjectOutputStream(fos);
4. oos.writeObject(myObject);
```

How do you fix the sample code above so a compressed version of myObject is saved?

a. Change the filename in line 2 to "myobject.zip", and then the FileOutputStream class will automatically add compression.
b. Replace line 3 with:
```
GZIPOutputStream zip = new GZIPOutputStream(fos);
ObjectOutputStream oos = new ObjectOutputStream(zip);
```
c. Replace line 4 with:
```
CompressedOutputStream cos = new (zip);
CompressedOutputStream(oos);
```
d. Replace line 2 with:
```
ZipFile zip = new ZipFile("myobject.ser");
FileOutputStream fos = new FileOutputStream(zip);
```
e. Replace line 3 with:
```
ZippedOutputStream zip = new ZippedOutputStream(fos);
FileOutputStream fos = new FileOutputStream(zip);
```

5. perm = new java.io.FilePermission(“/tmp/abc”,”read”);

What does the sample code above create when it is executed?

a. A FilePermission object representing the read access to the file named abc under the directory /tmp directory.
b. A security policy called perm, storing it in the FilePermission directory under /tmp/abc.
c. A new readable file in the perm method, storing it in the abc directory.
d. A Security Policy with the name FilePermission, reading it to the abc file.
e. A new file called Read, with the java.io.Filepermission set to default.

6. How do you save an encrypted version of a serialized object?

a. Have your class implement EncryptedSerializable, and implement the readEncryptedObject() and writeEncryptedObject() methods.
b. Save the serialized object to a StringBuffer, and encrypt the StringBuffer.
c. Pass the ObjectOutputStream argument of writeObject to a CipherWriter, and call writeObject() on the CipherWriter object to encrypt each field.
d. Generate a Cipher, and pass it to the SealedObject constructor along with the Serializable object.
e. Place the object in a JAR file, and set the isEncrypted() property for the JarEntry.

7. private static final
   ObjectStreamField[] serialPresistentFields = {
   new ObjectStreamField("brain", Point.class),
   new ObjectStreamField("bench", Dimension.class)
   };

   private Rectangle rect;

   private void readObject(ObjectInputStream ois)
        throws ClassNotFoundExeception, IOExeception {
   Point point = (Point)fields.get("brain",null);
   Dimension dimension = (Dimension)fields.get("bench",null);
   Rect = new Rectangle(point, dimension);
   }
   Based on the sample code above, which do you add to the class declaration to make it valid?
   a. getObjectStream = ObjectInputStream
   b. ObjectInputStream.readObject = GetFields
   c. ReadObject.GetObject = ObjectInputStream.readFields();
   d. Get.readObject = (Dimension)ois.readObject();
   e. ObjectInputStream.GetField fields = ois.readFields();

8. How do you count the number of lines in a text file?
   a. Ask a LineNumberInputStream after reading the whole file
   b. Ask a LineNumberReader after reading the whole file
   c. SubClass FilterInputStream and count the number of "\n" characters that go by.
   d. Read the file a line at a time via a BufferedOutputStream
   e. Ask a LineNumberReader before reading the whole file

9. ByteArrayOutputStream baos = new ByteArrayOutputStream();
   ObjectOutputStream out = new ObjectOutputStream(baos);
   Out.writeObject(new StringBuffer("Hello\uD801\uDFFE"));
   byte bArray[] = baos.toByteArray();
   In the sample code above, after execution, what does the array named bArray contain?
   a. A hash code created from StringBuffer
   b. The status of ByteArrayOutputStream baos(0x20 opened, 0x21 closed)
   c. A reference to ByteArrayOutputStream baos
   d. Unicode values for each character in Hello
   e. A serialized version of StringBuffer object containing string "Hello\uD801\uDFFE"