MISCELLENEOUS

1.

```
import javax.naming.*;
 public void createName() throws NamingException {
      Context context = new InitialContext();
      context.bind("/config/applicationBrain", "Bench");
 }
 public String getName() throws NamingException {
      Context context = new InitialContext();
      return (String) context.lookup("/config/applicationBrain");
 }
```

What function does the sample code above serve?

a. It looks for the file naming convention and returns it to the application.

b. It erases the JNDI context, creates a new object in the new context, and then retrieves an object from the context.

c. It establishes the JNDI context.

d. It opens the JNDI, inserts the naming convention, and stores it for later use.

e. It creates a new JNDI context every time it is run, while not storing any previous contexts

2.

```
DataSource dataSource = null;
try {
      Context context = new InitialContext();
      datasource = context.lookup("Database");
}
catch  (NamingException e){
 // Additional Code
}
```

What is the result of executing the sample code above?

a. It attempts to connect to the dataSource with the name "Database".

b. It sets the dataSource to null and renames its "Database".

c. It connects to the database called "DataSource" and renames it "e".

d. It deletes the old context and creates a new context with the name "Database".

e. The dataSource has an invalid name of Database.

3.

```
ProcessBuilder builder = new ProcessBuilder(javaBin, “-cp”,
            “-agentlib:jdwp=transport=dt_socket, server=y,
suspend=y, address=8000”, classpath, className);
builder.redirectErrorStream(true);
```

Based on the sample code above, which change do you make so an exception is not thrown?

a. The address in the classpath argument needs to be left blank
b. The classpath argument needs a value in the className field
c. The ProcessBuilder builder needs to be set to false.
d. The classpath argument must immediately follow the classpath value.
e. The classpath value must immediately follow the classpath argument.

4.

```
Static void PrintStuff(ClassDoc cd) {
      System.out.print(cd.modifiers());
      System.out.print(cd.isclass() ? “class” : “interface”);
      System.out.println(cd.qualifiedName());
      ClassDoc interfaces[] = cd. Interface();
      for (int i=0; n=interfaces.length; i<n; i++) {
      System.out.println(“ implements “ +
      interfaces[i].qualifiedName());
}

ClassdDoc parent = cd.superclass();
 if (parent != null ) {
      System.out.println(“ extends “ + parent.qualifiedName());
 }
}
```

Which ClassDoc refers to java.awt.Graphics2D in the above sample code, what output does the doclet generate?

a. public class java.awt.Graphics2D
   extends java.awt.Graphics
b. public abstract class Graphics2D
   extends Graphics
c. public interface Graphics2D
   extends Graphics
d. public abstract class java.awt.Graphics2D
   extends java.awt.Graphics
e. public interface java.awt.Graphics2D
   extends java.awt.Graphics

5. Which tool do you use to generate portable artifacts for web services?
    a. jws
    b. wsgen
    c. soap
    d. javac
    e. rmic

6. Sample code

   You properly import classes in the javax.xml.* packages and the definitions into the code:

   ```
   URL url = new URL("http://api.example.com/example.wsdl");
   QName=serviceName = new QName("urn:Example", "ExampleService");
   ```

   Based on the sample code above, which line of code do you use to create a web service for dispatching SOAP messages?
    a. Service.get(url, serviceName);
    b. Service.getInstance(url, serviceName);
    c. Service.create(url, serviceName);
    d. Service.get(url, serviceName, SoapMessage.class);
    e. Service.create(url, serviceName, Service.Mode.MESSAGE);

7. You are deploying an application via Java Web Start. The JAR files are signed. You want the client to get full system access when trusted by the user. Referring to the scenario above, which elements do you add to the JNLP file?
    a. <permission> <java.security.AllPermission/> </permission>
    b. <security> <permission=AllPermission/> </security>
    c. <permission> <all/> </permission>
    d. <security> <all-permissions/> </security>
    e. <security> <*/> </security>

8. How do you implement a custom permission in the Java security system?
    a. Submit a request for additional permission to the Java Community Process (JCP).
    b. Download the Java Source code from Sun, add the new permission to the default security manager, and recompile.
    c. Implement a new permission class, and add the permission to the security policy file.
    d. Define the new permission in the security policy file, and notify the HotSpot VM of the change.
    e. Create a new method to extend the Security properties, and add it to the rt.jar library.

9. You are trying to set up a Servlet Http session replication on a tomcat server with complex objects. Some of the objects implement serializable, but hold nonserializable members. The stacktraces are not providing any useful information.

   Based on the scenario above, how do you enable verbose class names in the stacktrace when a NotSerializableException occurs?

   a. Set the vm parameter debug.serializationerror.extended=true.
   b. Set the vm parameter sun.io.serialization.extendedDebugInfo=true.
   c. Set java.io.ObjectOutputStream.writeObject serialize to true.
   d. Set the vm parameter -XX:-UseSerialGC=true.
   e. Set Serialization.java.io.writeObject to true.

10. You are creating a Java App with EJBs. Some of the structure of the EJBs has been changed, and the old tables in the Java DB are not used anymore. The application deploys, but it no longer creates tables.

    Based on the scenario above, how do you resolve the table issue?

    a. Hard code the tables in the EJB.
    b. Create a jar with the table structure already complete.
    c. Restart the interface, and perform a clean build.
    d. Change the data source to match the jdbc.
    e. Rebuild the tables manually.

11.

```
import javax.script.*;
public class ExecuteScript {
  public static void main(String[] args) throws Exception {
     ScriptEngineManager factory= new ScriptEngineManager();
     ScriptEngine engine = factory.getEngineByName("JavaScript");
```

    Based on the sample code above, after discovering the JavaScript scripting engine, how does the code execute a block of JavaScript code from the Java program?

    a. Pass the code block to the eval() method of the engine.
    b. First pass the code to the the compile() method of the engine then, if it passes, call the evaluate() method.
    c. Pass the code block to the evaluate() method of the engine.
    d. Pass the code block to the execute() method of the engine.
    e. First pass the code to the the compile() method of the engine then, if it passes, call the execute() method.

12. Which class, found in the javax.script package, serves as the discovery mechanism for the installed set of scripting engines?
    a. ScriptEngineToolkit
    b. AbstractScriptEngine
    c. ScriptEngine
    d. ScriptEngineManager
    e. ScriptEngineFactory

13. Which one of the following Java APIs do you use to look up a Java MBean?
    a. JDBC
    b. Jini
    c. JMS
    d. JMAPI
    e. JNDI

14.
```
Line 1 for(int i=0; i<=22; ) {
Line 2     if (i<= 10) {
Line 3          int j = 2 + i;
Line 4          i++;
Line 5          // Do Something //
Line 6          System.out.println(“i: “ + i + “j: “ + j);
Line 7     }
Line 8 }
```

Which line of code do you add to complete the sample code above?
    a. class LoopingThread extends the Thread to the top of the code
    b. return new (valueBeforeIncrement); at the bottom of the code
    c. System.out.println(i+ “ “ +j);} at the bottom of the code
    d. Public static void loop(){ to the top of the code
    e. break; to the end of the loop

15. Which line of code do you use to replace Line 2 in the sample code above in order to print "Hello, World" when you pass just the class name to the Java command?
    a. public static main void(String []args{}) {
    b. static public void main(String args[*]) {
    c. public static void main(String args...) {
    d. public static main void(String[] args) {
    e. public static void main(String... args) {