

Chapter 12 – I/O Streams

Question 1:

In order to denote a variable that should not be written out as part of its class' persistent state the modifier to be used is

- A) private
- B) protected
- C) persistent
- D) transient
- E) static

Question 2:

True or False:

The method `readInt` does not exist in classes that extend `Reader`.

- A) True
- B) False

Question 3:

True or False:

If an instance of the class File is created in order to represent a file, which doesn't exist, the file will be created.

- A) True
- B) False

Question 4:

True or False:

The RandomAccessFile class doesn't belong to the classes hierarchy Reader\Writer nor to the classes hierarchy InputStream\OutputStream.

- A) True
- B) False

Question 5:

True or False:

The RandomAccessFile class has the necessary methods to read numerical values from a file.

- A) True
- B) False

Question 6:

True or False:

The UTF character needs 16 bits.

- A) True
- B) False

Question 7:

True or False:

Constructing an instance of File doesn't create a file on local file system.

- A) True
- B) False

Question 8:

True or False:

In creating an instance from the RandomAccessFile class the mode string can be either "r" or "rw".

- A) True
- B) False

Question 9:

Given the code below,

```
RandomAccessFile raf = new RandomAccessFile(myFileName, "rw");
```

Which of the following sentences (one or more) is true ?

- () A) If the file which its name is myFileName doesn't exist then a zero-length file is created,
- () B) If the file which its name is myFileName doesn't exist then a FileNotFoundException is thrown.

Question 10:

Given the code below,

```
RandomAccessFile raf = new RandomAccessFile(myFileName, "r");
```

Which of the following sentences (one or more) is true ?

- () A) If the file which its name is myFileName doesn't exist then a zero-length file is created,
- () B) If the file which its name is myFileName doesn't exist then a FileNotFoundException is thrown.

Question 11:

True or False:

In order to read 100 integer numbers from a file in which they are stored it is possible to use the FileReader class and its method readInt().

- A) True
- B) False

Question 12:

What is the output of the following code ?

```
1.  try
2.  {
3.      RandomAccessFile raf1 = new RandomAccessFile("dataFile","rw");
4.      BufferedOutputStream bos = new BufferedOutputStream(raf1);
5.      DataOutputStream dos = new DataOutputStream(bos);
6.      dos.writeInt(2000);
7.      dos.close();
8.      bos.close();
9.      raf.close();
10.     RandomAccessFile raf2 = new RandomAccessFile("dataFile","r");
11.     BufferedInputStream bis = new BufferedInputStream(raf2);
12.     DataInputStream dis = new DataInputStream(bis);
13.     System.out.println("The year is " + dis.readInt());
14.     dis.close();
15.     bis.close();
16.     raf2.close();
17. }
18. catch(Exception e)
```

19. {}

- () A) The output will be: "2000".
- () B) The output will be: "The year is 2000".
- () C) The output will be: "The year is".
- () D) The code will not compile.
- () E) The code will compile successfully but during execution an exception will be thrown.

Question 13:

Given the code below:

```
1.  import java.io.*;
2.  class Neptun implements Serializable
3.  {
4.      String name;
5.      transient ObjectOutputStream oos;
6.      transient ObjectInputStream ois;
7.      Neptun()
8.      {
9.      }
10.     Neptun(String str)
11.     {
12.         name = str;
13.     }
```

```
14.     public static void main(String args[])
15.     {
16.         Neptun neptun = new Neptun();
17.         neptun.go();
18.     }
19.     public void go()
20.     {
21.         try
22.         {
23.             oos = new ObjectOutputStream(new FileOutputStream("stam.txt"));
24.             oos.writeObject(new Neptun("jojo"));
25.             oos.close();
26.             ois = new ObjectInputStream(new FileInputStream("stam.txt"));
27.             Neptun neptun = (Neptun)ois.readObject();
28.             System.out.println(neptun.name);
29.         }
30.         catch(Exception e){}
31.     }
32. }
```

- A) If the word transient is removed from the program then the code won't run successfully
- B) The output is "jojo"
- C) It is necessary to do casting in row 27
- D) The try& catch sentence is necessary
- E) The ois and oos instance variables are not written to the file

Question 14:

True or False:

When writing an object, which is the head of a linked list, all of the elements that belongs to that list will be written too unless the 'next' variable was marked as transient.

- A) True
- B) False

Question 15:

True or False:

The modifier to be used in order to denote a variable that should not be written out as part of its class' persistent state is "transient".

- A) True
- B) False

Question 16:

True or False:

When writing an object, which is the head of a binary tree, all of the elements that belong to that binary tree will be written too unless the 'right' and 'left' variables were marked (both of them or only one of them) as transient.

- A) True
- B) False

Question 17:

True or False:

When instantiating the class `FileInputStream` using a file that doesn't exist, a new file will be created.

- A) True
- B) False

Question 18:

True or False:

When instantiating the class `FileOutputStream` using a file that doesn't exist, a new file might be created.

- A) True
- B) False

Question 19:

True or False:

The `Externalizable` interface extends the `Serializable` interface and its purpose is to give the programmer the control over how to serialize an object. When we write an object into an `ObjectOutputStream` using the `writeObject` method and the object was instantiated from a class that implements `Externalizable` then the default serialization mechanism doesn't take place. Instead, The methods `writeExternal` and `readExternal` are invoked on the serialized object. When a given class implements the `Externalizable` interface the serialization process is no longer recursive and the given class is responsible for making the decision regarding the way in which the object members (as well as super class members) are serialized.

- A) True
- B) False

Question 20:

True or False:

The Unicode is a character coding system designed to support the interchange, processing and display of the written texts of the diverse languages of the modern world. The system property called `file.encoding` determines the type of the Unicode encoding the system uses. Like other system properties, this system property can be checked by calling `System.getProperty("file.encoding")`.

- A) True
- B) False

Question 21:

Given the code below:

```
1. import java.io.*;
2.
3. class Uti
4. {
5.     static final int SIZE = 10000;
6.     static final byte[] buffer = new byte[SIZE];
7.
8.     public static void main(String args[])throws IOException
9.     {
10.         copy(args[0], args[1]);
```

```
11.     }
12.
13.     static void copy(String src, String dst) throws IOException
14.     {
15.         InputStream fis = null;
16.         OutputStream fos = null;
17.         try
18.         {
19.             fis = new FileInputStream(src);
20.             fos = new FileOutputStream(dst);
21.             while(true)
22.             {
23.                 synchronized(buffer)
24.                 {
25.                     if(fis.read(buffer)==-1)
26.                     {
27.                         break;
28.                     }
29.                     fos.write(buffer);
30.                 }
31.             }
32.         }
33.         finally
34.         {
35.             if(fis!=null)
```

```
36.      {
37.          fis.close();
38.      }
39.      if(fos!=null)
40.      {
41.          fos.close();
42.      }
43.  }
44.  }
45. }
```

- () A) The code doesn't compile.
- () B) The needed change in this program, so it will compile successfully is adding a catch block.
- () C) The code compile successfully.
- () D) The fis and fos variables' type must be FileInputStream and FileOutputStream.