

Chapter 12 – I/O Streams

Answers

1. D
Only variables can have the modifier transient. A main reason to mark a variable as transient is so it will not be written out through a stream.
2. A
Reader classes deal with character streams and were created to work with Unicode characters. These classes have the read() method, but not the readInt() method.
3. B
An instance of the File class does not create an actual file. An instance of a File object represents a file or directory, and is used to view information on the file or directory represented.
4. A
The RandomAccessFile class extends directly from the Object class. The RandomAccessFile class can be used to read or modify a file, and it is not compatible with the Reader and Writer classes and the InputStream and OutputStream classes.
5. A
The RandomAccessFile class has methods to read and write the primitive numerical data types. For example it has the methods “readInt” and “writeInt” to read and write primitive int values to and from a file.
6. B
UTF encoding uses as many bytes as is needed to encode a specific character.

7. A

An instance of the File class does not create an actual file.

8. A

When the mode passed is "r" the file is opened for reading only. When the mode passed is "rw" the file is opened for reading and writing.

9. A

When a RandomAccessFile object is created, if the mode passed is "r" for read only, then if the file does not exist a FileNotFoundException is thrown. If the mode passed is "rw" for read/write then if the file does not exist a new file with a zero length is created, and this is the case presented here.

10. B

When a RandomAccessFile object is created, if the mode passed is "r" for read only, then if the file does not exist a FileNotFoundException is thrown. This is the case in the example here. If the mode passed were to be "rw" for read/write then if the file did not exist a new file with a zero length would be created.

11. B

The FileReader class does not have the "readInt" method. It just has a "read" method.

12. D

The code will not compile since on line 4 when creating a BufferedOutputStream instance, the constructor needs to be passed an object of type OutputStream, and a RandomAccessFile object is not a type of OutputStream. The same type rule applies on line 11 when creating a BufferedInputStream instance, it needs to be passed an InputStream object.

13. A,B,C,D,E

In this example a new ObjectOutputStream object is created, and it is

passed a new `FileOutputStream` as a parameter. When the `FileOutputStream` is created it is passed the name of a file to create (or to use if it already exists), with the name "stam.txt".

Then a new `Neptun` object is created, and it is passed to the "writeObject" method of the `ObjectOutputStream`. A representation of the `Neptun` object is written to the file. The `String` variable called "name" is written to the file created, whereas the information in the "oos" and "ois" variables is not written, since they are marked with the modifier "transient". The `FileOutputStream` object is then set to null.

After this a `ObjectInputStream` object is created, and it is passed a new `FileInputStream` object, which uses the same "stam.txt" file. The "readObject" method is then called on this `ObjectInputStream`, and the object that was represented in the file is returned. Since the "readObject" returns a reference of type `Object`, and we are putting it into a variable that holds a reference of type `Neptun`, a cast is needed here. After this the value in the "name" parameter of this `Neptun` object is printed, which says "jojo". The try and catch block here is necessary since most of the functionality having to do with streams can throw some sort of exception.

14. A

Each object in a linked list has a variable that holds a reference to the next object in the list, lets say called "next". If the first object in the list is written to a stream, the information for the other objects will be written to the stream as well, as long as these "next" variables are not marked as transient. Once one of the "next" variables is marked as transient, the rest of the objects in the list will not be written to the stream.

15. A

Only variables can have the modifier transient. A main reason to mark a variable as transient is so it will not be written out through a stream.

16. A

This statement is true.

17. B

When trying to create an instance of a `FileInputStream`, if the filename passed does not exist a `FileNotFoundException` will be thrown.

18. A

An instance of a `FileOutputStream` can be created to actually create a new file on the computer. For example if you write the following statement:

```
FileOutputStream fis = new FileOutputStream("test.txt");
```

If the file `test.txt` does not exist, then it will be created.

19. A

These statements are correct.

20. A

These statements are correct.

21. C

This code compiles fine, but since there is no catch block, if the file in the "src" variable that is passed to the `FileInputStream` does not exist a `FileNotFoundException` would occur during runtime. If there was a catch block then this exception would be caught and the program would continue.