

Critical Response Questions – Chapter 11

1. Can data in memory be called a file? Explain.

No. Data in memory exists temporarily in RAM, while a file is a permanent collection of data stored on a secondary storage device such as a hard drive or SSD. Memory contents are volatile and disappear when the program ends, while files persist until deleted.

2. Write the import statement required to access the File Class in an application.

```
import java.io.File;
```

3. Identify the error in the following statement:

```
File textFile = new File("c:\inventory.txt");
```

The backslash \ is an escape character in Java strings. It must be written as a double backslash.

Correct version:

```
File textFile = new File("c:\\inventory.txt");
```

4. a) Which statement is used to write an exception handler?

The try-catch statement.

4. b) Write an exception handler to handle an IOException if a specified file name cannot be used to create a file. The exception handler should display appropriate messages to the user.

```
try {  
    PrintWriter pw = new PrintWriter("data.txt");  
} catch (IOException e) {  
    System.out.println("Error: The file could not be created.");  
    System.out.println("Details: " + e.getMessage());  
}
```

5. a) What is the name of the stream for displaying error messages?

System.err

5. b) Where are these messages displayed?

They are displayed on the standard error output, typically the console or terminal window.

6. a) What does the file stream keep track of?

A file stream keeps track of the current read/write position within the file.

6. b) What characters together make up a line terminator?

A line terminator typically consists of:

- \r\n in Windows
- \n in Unix/Linux/macOS
- \r historically used in older Mac systems

7. What two classes are used together to write data to a file?

Commonly:

- FileWriter and PrintWriter
Or:
- FileOutputStream and DataOutputStream

8. Write a statement to convert account balances that have been read from a text file to a double value and add them to totalBalance.

```
totalBalance += Double.parseDouble(balanceString);
```

9. Explain the difference between object serialization and object deserialization.

Serialization is the process of converting an object into a byte stream so it can be saved to a file or transferred over a network.

Deserialization is the reverse process: converting the byte stream back into a fully constructed Java object.

10. What interface must be implemented if objects of a class are to be written to a file?

java.io.Serializable