Review

Challenge #1 — Scored

Your Answers | Your Notes (Editable)

1. i. What is java I/O?
   ii. Explain what is java Writer class and list all the sub-classes of the writer class.

   1. In Java, a Writer is an abstract class used for writing character streams. It is the superclass of all classes that write characters to an output stream.

   2. A java writer class is an abstract class in java that is used for writing to character streams. It is the superclass of all classes representing character output streams.

   SUBCLASSES OF WRITER CLASS;
   BufferedWriter - Writes text to a character-output stream, buffering characters so as to provide for the efficient writing of single characters, arrays, and strings.
   CharArrayWriter - A class that can be used to write characters to an array.
   FilterWriter - A class that implements filtering writing.
   OutputStreamWriter - A class that connects character streams to byte streams.
   FileWriter - A class that writes text to a file in the default encoding.
   StringWriter - A class that writes characters to a string buffer.
   PrintWriter - A class that prints text-output stream.

2. i. Explain java Reader class and list all the sub-classes of the Reader class.
ii. Explain the java FileWriter class.

1. Java Reader class is an abstract class in Java that is used for reading character streams. It is the superclass of all classes representing character input streams.

SUBCLASSES OF READER CLASS;
BufferedReader
CharArrayReader
FilterReader
InputStreamReader
PipedReader
StringReader

2. The Java FileWriter class is a subclass of the Writer class and is used for writing character streams. It is used to write characters to a file. It provides methods for writing characters, strings, and arrays of characters to a file. The FileWriter class is used to create a file if it does not exist, and if the file already exists, it will overwrite the existing content.

3. i. What is java BufferedReader class
ii. Explain what you understand by Buffering.

1. The Java BufferedReader class is used to read text from a character-input stream, buffering characters so as to provide for the efficient reading of characters, arrays, and lines.

2. Buffering is a method used to temporarily store data in memory before writing it to a file or sending it over a network. This can improve performance by reducing the number of read and write operations, as data is read or written in larger chunks rather than one character at a time.

4. i. Explain java File class.
ii. List the tasks you can perform using the java File class.
iii. What is the use of the ByteArrayOutputStream class?

1. The Java File class is used to handle files and directories. It provides methods for creating, deleting, renaming, and querying files and directories on the file system.

2. Tasks you can perform using the Java File class include:

* Creating new files and directories
* Deleting files and directories
* Renaming files and directories
* Checking if a file or directory exists
* Listing the contents of a directory
* Getting information about a file or directory, such as its size, path, and last modified timestamp

3. The ByteArrayOutputStream class in Java is a subclass of OutputStream that creates an in-memory output stream.

5. i. What is the difference between the throw and throws keyword.
ii. What is java Exception? list 2 types
iii. what type of Exception is used to handle FileReader and FileWriter.

1. The difference between a throw and throws keyword, is that throw is defined inside a class method while throws is defined in the method's argument (Just before the curly bracket). Throws is used to catch defined error within the method while throws is used to catch all possible exception errors in the class method.

2. Checked Exception
Unchecked Exception

3. The type of exception used to handle FileReader and FileWriter is Unchecked Exception.

6. Using the FileWriter class, write this text "I have successfully written into a csv file", into your desktop. kindly show your work using code, remember to print your CSV file to the desktop.

```
public static void main(String[] args) {
    String text = "i have successfully written into a csv file";
    String desktopPath = System.getProperty("user.home") + "/Desktop/";
    String fileName = "output.csv";
    try {
        FileWriter writer = new FileWriter(desktopPath + fileName);
        writer.write(text);
        writer.close();
        System.out.println("File written successfully to: " + desktopPath +
fileName);
    } catch (IOException e) {
        System.out.println("An error occurred.");
    }
}
```

CODE STEPS;
1. Defined variables (File name, File content and file saving path)
2. Initiate try
3. Initialize FileWriter
4. Write text to File
5. Close FileWriter
6. Sout success
7. Catch try exceptions (Sout failure)

7. i. Using the FileReader class read the text from question 6 into your console, kindly show your work using code.

```
public static void main(String[] args) {
    String desktopPath = System.getProperty("user.home") + "/Desktop/";
    String fileName = "output.csv";
    try {
        BufferedReader reader = new BufferedReader(new
FileReader(desktopPath + fileName));
        String line = reader.readLine();
        System.out.println(line);
        reader.close();
    } catch (IOException e) {
        System.out.println("An error occurred.");
    }
}
```

CODE STEPS;
1. Defined variables (File name and file path)
2. Initiate try
3. Initialize BufferReader
4.
5.
6.
7.

Your Comments

Optional comments about the assessment process, challenges, etc.

You can no longer make changes to your challenge submissions. You can manage your account settings or log out if you are done.

Log Out