**Solution: Writing data to a file**

1. Introduce Groovy and explain why it is a good language for working with files.

Groovy is a dynamic, object-oriented programming language for the Java platform. It is both a static-typed and dynamic-typed language, offering flexibility and a concise syntax. Groovy simplifies many common tasks, making it an excellent choice for working with files. Groovy's built-in File I/O capabilities allow for easy reading, writing, and processing of files. The language also seamlessly integrates with Java, allowing developers to leverage existing Java libraries and code.

1. Sample data set to be written to a file:

def sampleData = [

['John Doe', 'johndoe@example.com', 'Male', '31'],

['Jane Doe', 'janedoe@example.com', 'Female', '28'],

['Bob Smith', 'bobsmith@example.com', 'Male', '45'],

['Sara Johnson', 'sarajohnson@example.com', 'Female', '22']

]

1. & 4. Groovy script to create a file and write the data set to it:

// Function to write data to a CSV file

def writeCSV(String filename, List<List<String>> data) {

try {

File file = new File(filename)

file.withWriter('UTF-8') { writer ->

// Write the header line

writer.writeLine 'Name, Email, Gender, Age'

// Write the data

data.each { row ->

writer.writeLine row.join(', ')

}

}

} catch (IOException e) {

println "Error writing to file: ${e.message}"

}

}

// Main script

def sampleData = [

['John Doe', 'johndoe@example.com', 'Male', '31'],

['Jane Doe', 'janedoe@example.com', 'Female', '28'],

['Bob Smith', 'bobsmith@example.com', 'Male', '45'],

['Sara Johnson', 'sarajohnson@example.com', 'Female', '22']

]

writeCSV('data.txt', sampleData)

1. Students should test their scripts with different data sets, ensuring flexibility and adaptability to various scenarios.
2. The provided script is documented and explains how it works.
3. Students can use the provided script to create and write sample data sets to a file.
4. Students should reflect on what they've learned and consider how they can apply this new knowledge to real-world projects. This can include creating applications that require file processing, such as data analysis tools, or integrating with existing Java projects that require file handling.