**Solution: Reading data from 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 file containing data in CSV format:

Name, Email, Gender, Age

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 read data from the file and process it:

import groovy.json.JsonOutput

// Define a class to represent a person

class Person {

String name

String email

String gender

int age

}

// Function to read data from a CSV file and return a list of Person objects

def readCSV(String filename) {

List<Person> people = []

try {

File file = new File(filename)

List<String> lines = file.readLines()

// Remove the header line

lines.remove(0)

// Process each line

lines.each { line ->

List<String> fields = line.split(',\\s\*')

people << new Person(name: fields[0], email: fields[1], gender: fields[2], age: fields[3] as int)

}

} catch (IOException e) {

println "Error reading file: ${e.message}"

}

return people

}

// Main script

def people = readCSV('data.csv')

println JsonOutput.prettyPrint(JsonOutput.toJson(people))

1. Students should test their scripts with different data files, 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 read and process the sample data file (data.csv).
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.