

**JAVA SOFTWARE DEVELOPER – ANESTHESIA INTERVIEW PROMPT**

Attached is a text file containing a valid JSON structure which models a family tree.

**JSON Specifications:**

1. The JSON Structure lists the following for each person:
   1. A unique ID number
   2. A name
   3. An ID number identifying the person’s spouse
   4. A list of ID numbers identifying a person’s children
2. A person may have any number of children (greater than or equal to 0).
3. Not all persons have a spouse; these individuals have a Spouse ID of **-1**.
4. Not all persons have a parent included in the JSON Structure.
5. The JSON file may have any number of persons, not all of which are necessarily connected to any or all other persons.

**Sample of JSON:**

[… {"id": 11,"name": "Janet Brennan","spouseid": 12,"children": [13, 14, 15]}, {"id": 12,"name": "Peters Mcdaniel","spouseid": 11,"children": [13,14,15]}, ...]

**Task:**

Write a Java program to answer the question: How many grandchildren are in the JSON Structure?

**Additional Specifications:**

A person is considered a grandchild if he or she is the child of another person’s child, as defined in the JSON Family Tree. A person is also considered a grandchild if he or she is the spouse of a grandchild.

**Requirements:**

Your program should read and parse the JSON text file, analyze the data, and output an answer to the above question. Additionally, your submission should be built in such a way that we can easily run the program. Please be sure to include all source code, as we will review your implementation. Your implementation should demonstrate your Object-Oriented Design skills and should otherwise be attentive to extensibility and efficiency.

We expect you to complete the problem on your own, but feel free to utilize any resources (external libraries, tools, etc.) that you feel will better your solution – be prepared to explain why you chose to leverage such tools. Finally, be prepared to provide an explanation of your high level approach to the problem, discuss your design process, and defend your implementation.