**JSON and Collections**

**72 hour grading deadline**

**Testing (cont.)**

**Generically speaking: Java Assert();**

<https://www.geeksforgeeks.org/assertions-in-java/>

**Testing Example code:**

<https://www.java2novice.com/junit-examples/>

<https://www.vogella.com/tutorials/JUnit/article.html>

<https://www.java2novice.com/junit-examples/test-annotation/>

**JSON, Collections (slides)**

Important Points:

Slide: “Parsing JSON in Java”

**Important:**

1) build classes with fields for the desired elements of the JSON

2) use the same names

3) get the types right

Slide: “Lists”

 “Ordered” not “sorted”

**A word about Design:**

* Decoupling data from code and io streams
* Data objects

**Visual Representation of Linked List**

[**https://visualgo.net/en/list**](https://visualgo.net/en/list)

**Adventure.pdf**

**JSON Schema**

This is the “schema”. This is NOT the same as a JSON document.

Direction := {

directionName : String

room : String //name of the room the direction is point to

}

Room := {

name : String

description : String

directions : Direction []

}

Layout := {

startingRoom : String //the name of the starting room

endingRoom : String //the name of the ending room

rooms : Room []

}

**JSON Document:**

{

 "ID": "1",

 "First\_Name": "Shikhar",

 "Last\_Name": "Dhawan",

 "Date\_Of\_Birth": "1981-12-05",

 "Place\_Of\_Birth":"Delhi",

 "Country": "India"

}

Reading JSON in Java

<https://www.tutorialspoint.com/how-to-read-the-contents-of-a-json-file-using-java>

See section “Decoding JSON in Java”

<https://www.tutorialspoint.com/json/json_java_example.htm>

<http://www.jsonschema2pojo.org/>

<https://www.geeksforgeeks.org/parse-json-java/>

 How to download a file from a URL in Java

<https://www.journaldev.com/924/java-download-file-url>