**Julio Pochet** - **Module 11 Discussion Board**:

**Understanding JSON and Parsing in Java**

JSON (JavaScript Object Notation) is a lightweight, text-based format designed to store and exchange data in a structured way. It represents data as key-value pairs and arrays, making it straightforward for both humans and machines to read. While it originated from JavaScript, JSON is language-independent and widely used in web services, APIs, and data storage. A simple example of a JSON object looks like this:

{ "name": "Alice", "age": 28, "skills": ["Java", "SQL"] }

Its clear syntax and universal support across platforms make it a standard format for modern applications (MDN Web Docs, 2024).

In Java, parsing JSON involves converting JSON text into Java objects or retrieving data from a JSON structure. Several libraries can accomplish this, including **json-simple**, **Gson**, and **Jackson**. Using the json-simple library, developers can parse a JSON file or string using the JSONParser class and extract data through methods like get(). For example:

JSONParser parser = new JSONParser();

Object obj = parser.parse(new FileReader("data.json"));

JSONObject jsonObj = (JSONObject) obj;

String name = (String) jsonObj.get("name");

More advanced libraries, like Gson and Jackson, allow direct mapping of JSON data into Java classes, which is especially useful for large or nested datasets (Baeldung, 2023).

**References**  
Baeldung. (2023). Introduction to JSON in Java. <https://www.baeldung.com/java-json>  
MDN Web Docs. (2024). Working with JSON. <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>  
GeeksforGeeks. (2024). Working with JSON data in Java. <https://www.geeksforgeeks.org/java/working-with-json-data-in-java/>