📌 Practice Problems

1️⃣Create a JSON object for a Student with fields: name, age, and subjects (array).

```java
import org.json.JSONArray;
import org.json.JSONObject;

public class StudentJson {
    public static void main(String[] args) {
        JSONObject student = new JSONObject();
        student.put("name", "ABC");
        student.put("age", 20);

        JSONArray subjects = new JSONArray();
        subjects.put("Math");
        subjects.put("Science");
        subjects.put("History");

        student.put("subjects", subjects);
        System.out.println(student.toString());
    }
}
```

2️⃣Convert a Java object (Car) into JSON format.

```java
import com.fasterxml.jackson.databind.ObjectMapper;

class Car {
    public String brand;
    public String model;
    public int year;
    public Car(String brand, String model, int year) {
        this.brand = brand;
        this.model = model;
        this.year = year;
    }
}

public class CarJson {
    public static void main(String[] args) throws Exception {
        ObjectMapper objectMapper = new ObjectMapper();
        Car car = new Car("XYZ", "Sedan", 2022);
        String json = objectMapper.writeValueAsString(car);
        System.out.println(json);
    }
}
```

3 **Read a JSON file and extract only specific fields (e.g., name, email).**

```
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import java.io.File;

public class ReadJsonFields {
    public static void main(String[] args) throws Exception {
        ObjectMapper objectMapper = new ObjectMapper();
        JsonNode jsonNode = objectMapper.readTree(new File("data.json"));
        System.out.println("Name: " + jsonNode.get("name").asText());
        System.out.println("Email: " + jsonNode.get("email").asText());
    }
}
```

4️⃣ Merge two JSON objects into one.

```java
import org.json.JSONObject;

public class MergeJson {
    public static void main(String[] args) {
        JSONObject obj1 = new JSONObject();
        obj1.put("name", "ABC");
        obj1.put("age", 25);

        JSONObject obj2 = new JSONObject();
        obj2.put("email", "abc@example.com");
        obj2.put("city", "XYZ");

        JSONObject merged = new JSONObject(obj1, JSONObject.getNames(obj1));
        for (String key : JSONObject.getNames(obj2)) {
            merged.put(key, obj2.get(key));
        }

        System.out.println(merged.toString());
    }
}
```

5️⃣ Validate JSON structure using Jackson.

```java
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.github.fge.jsonschema.core.exceptions.ProcessingException;
import com.github.fge.jsonschema.main.JsonSchema;
import com.github.fge.jsonschema.main.JsonSchemaFactory;
import java.io.File;
```

```
public class ValidateJson {
    public static void main(String[] args) throws Exception {
        ObjectMapper objectMapper = new ObjectMapper();
        JsonNode schemaNode = objectMapper.readTree(new File("schema.json"));
        JsonNode jsonData = objectMapper.readTree(new File("data.json"));
        JsonSchema schema =
JsonSchemaFactory.byDefault().getJsonSchema(schemaNode);
        System.out.println(schema.validate(jsonData).isSuccess() ? "Valid JSON"
: "Invalid JSON");
    }
}
```

6 Convert a list of Java objects into a JSON array.

```
import com.fasterxml.jackson.databind.ObjectMapper;
import java.util.Arrays;
import java.util.List;

class Person {
    public String name;
    public int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
}
```

```
public class ListToJson {
    public static void main(String[] args) throws Exception {
        List<Person> people = Arrays.asList(new Person("ABC", 22), new
Person("PQR", 30));
        String json = new ObjectMapper().writeValueAsString(people);
        System.out.println(json);
    }
}
```

7 **Parse JSON and filter only those records where age > 25.**

```
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import java.io.File;
import java.util.Iterator;

public class FilterJson {
    public static void main(String[] args) throws Exception {
        ObjectMapper objectMapper = new ObjectMapper();
        JsonNode root = objectMapper.readTree(new File("data.json"));
        Iterator<JsonNode> elements = root.elements();

        while (elements.hasNext()) {
            JsonNode person = elements.next();
            if (person.get("age").asInt() > 25) {
                System.out.println(person.toString());
            }
        }
    }
}
```

Hands-on Practice Problems

1️⃣ Read a JSON file and print all keys and values.

```java
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import java.io.File;
import java.util.Iterator;

public class ReadJsonKeysValues {
    public static void main(String[] args) throws Exception {
        ObjectMapper objectMapper = new ObjectMapper();
        JsonNode root = objectMapper.readTree(new File("data.json"));

        Iterator<String> fieldNames = root.fieldNames();
        while (fieldNames.hasNext()) {
            String field = fieldNames.next();
            System.out.println(field + ": " + root.get(field));
        }
    }
}
```

2️⃣ Convert a list of Java objects into a JSON array.

```java
import com.fasterxml.jackson.databind.ObjectMapper;
import java.util.Arrays;
import java.util.List;

class Employee {
    public String name;
    public int age;

    public Employee(String name, int age) {
        this.name = name;
        this.age = age;
    }
}
```

```java
public class ListToJsonArray {
    public static void main(String[] args) throws Exception {
        List<Employee> employees = Arrays.asList(new Employee("ABC", 25), new
Employee("PQR", 28));
        String json = new ObjectMapper().writeValueAsString(employees);
        System.out.println(json);
    }
}
```

3️⃣ **Filter JSON data: Print only users older than 25 years.**

```java
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import java.io.File;
import java.util.Iterator;

public class FilterUsersByAge {
    public static void main(String[] args) throws Exception {
        ObjectMapper objectMapper = new ObjectMapper();
        JsonNode root = objectMapper.readTree(new File("users.json"));

        Iterator<JsonNode> elements = root.elements();
        while (elements.hasNext()) {
            JsonNode user = elements.next();
            if (user.get("age").asInt() > 25) {
                System.out.println(user.toString());
            }
        }
    }
}
```

4️⃣ Validate an email field using JSON Schema.

```java
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.github.fge.jsonschema.core.exceptions.ProcessingException;
import com.github.fge.jsonschema.main.JsonSchema;
import com.github.fge.jsonschema.main.JsonSchemaFactory;
import java.io.File;

public class ValidateEmailJson {
    public static void main(String[] args) throws Exception {
        ObjectMapper objectMapper = new ObjectMapper();
        JsonNode schemaNode = objectMapper.readTree(new
File("email_schema.json"));
        JsonNode jsonData = objectMapper.readTree(new File("data.json"));
        JsonSchema schema =
JsonSchemaFactory.byDefault().getJsonSchema(schemaNode);
        System.out.println(schema.validate(jsonData).isSuccess() ? "Valid JSON"
: "Invalid JSON");
    }
}
```

5️⃣ Merge two JSON files into a single JSON object.

```java
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import java.io.File;

public class MergeJsonFiles {
    public static void main(String[] args) throws Exception {
        ObjectMapper objectMapper = new ObjectMapper();
        JsonNode json1 = objectMapper.readTree(new File("file1.json"));
        JsonNode json2 = objectMapper.readTree(new File("file2.json"));

        ((ObjectNode) json1).setAll((ObjectNode) json2);
        System.out.println(json1.toString());
    }
}
```

6️⃣ Convert JSON to XML format.

```java
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.dataformat.xml.XmlMapper;
import java.io.File;

public class JsonToXml {
    public static void main(String[] args) throws Exception {
        ObjectMapper objectMapper = new ObjectMapper();
        JsonNode jsonNode = objectMapper.readTree(new File("data.json"));
        String xml = new XmlMapper().writeValueAsString(jsonNode);
        System.out.println(xml);
    }
}
```

7️⃣ Convert CSV data into JSON.

```java
import com.fasterxml.jackson.databind.MappingIterator;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.dataformat.csv.CsvMapper;
import com.fasterxml.jackson.dataformat.csv.CsvSchema;
import java.io.File;
import java.util.List;

public class CsvToJson {
    public static void main(String[] args) throws Exception {
        File csvFile = new File("data.csv");
        CsvMapper csvMapper = new CsvMapper();
        CsvSchema schema = CsvSchema.emptySchema().withHeader();
        MappingIterator<Object> it =
csvMapper.readerFor(Object.class).with(schema).readValues(csvFile);

        List<Object> data = it.readAll();
        String json = new ObjectMapper().writeValueAsString(data);
        System.out.println(json);
    }
}
```

8️⃣ Generate a JSON report from database records.

```
import com.fasterxml.jackson.databind.ObjectMapper;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

class Record {
    public int id;
    public String name;
    public int age;

    public Record(int id, String name, int age) {
        this.id = id;
        this.name = name;
        this.age = age;
    }
}

public class DatabaseToJson {
    public static void main(String[] args) throws Exception {
        Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/db", "user", "pass");
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery("SELECT * FROM users");

        List<Record> records = new ArrayList<>();
        while (rs.next()) {
            records.add(new Record(rs.getInt("id"), rs.getString("name"),
rs.getInt("age")));
        }

        String json = new ObjectMapper().writeValueAsString(records);
        System.out.println(json);

        conn.close();
    }
}
```

Problem Statement: IPL and Censor Analyzer

Objective:

Develop a **Java application** that reads IPL match data from **JSON and CSV files**, processes the data based on defined **censorship rules**, and writes the sanitized data back to new files.

📌 Requirements

1️⃣Input Data Formats

The application should support:

- **JSON Input:** IPL match data in JSON format.
- **CSV Input:** IPL match data in CSV format.

2️⃣Censorship Rules

The program should apply the following censorship:

- **Mask Team Names:** Replace part of the team name with `"***"`.
  Example: `"Mumbai Indians"` → `"Mumbai ***"`
- **Redact Player of the Match:** Replace player names with `"REDACTED"`.

3️⃣Output Data Formats

- Generate **censored JSON and CSV files** after processing.

📝 Sample IPL Data

JSON Input (Before Censorship)

```
[
  {
    "match_id": 101,
    "team1": "Mumbai Indians",
    "team2": "Chennai Super Kings",
    "score": {
      "Mumbai Indians": 178,
      "Chennai Super Kings": 182
    },
    "winner": "Chennai Super Kings",
    "player_of_match": "MS Dhoni"
  },
  {
    "match_id": 102,
    "team1": "Royal Challengers Bangalore",
    "team2": "Delhi Capitals",

    "score": {

      "Royal Challengers Bangalore": 200,

      "Delhi Capitals": 190

    },

    "winner": "Royal Challengers Bangalore",

    "player_of_match": "Virat Kohli"

  }

]
```

CSV Input (Before Censorship)

```
match_id,team1,team2,score_team1,score_team2,winner,player_of_match

101,Mumbai Indians,Chennai Super Kings,178,182,Chennai Super Kings,MS
Dhoni

102,Royal Challengers Bangalore,Delhi Capitals,200,190,Royal
Challengers Bangalore,Virat Kohli
```

```
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.node.ArrayNode;
import com.fasterxml.jackson.databind.node.ObjectNode;
import java.io.File;

public class IplCensor {
    public static void main(String[] args) throws Exception {
        ObjectMapper objectMapper = new ObjectMapper();
        ArrayNode matches = (ArrayNode) objectMapper.readTree(new
File("ipl.json"));

        for (JsonNode match : matches) {
            ((ObjectNode) match).put("team1",
censorTeam(match.get("team1").asText()));
            ((ObjectNode) match).put("team2",
censorTeam(match.get("team2").asText()));
            ((ObjectNode) match).put("player_of_match", "REDACTED");
        }

        objectMapper.writeValue(new File("ipl_censored.json"), matches);
        System.out.println("Censored JSON saved.");
    }

    private static String censorTeam(String team) {
        return team.split(" ")[0] + " ***";
    }
}
```