Page No.
Date

Assignment - B6

* Date of Completition: 25/11/2020

* Date of Submission: 27/11/2020

* Title: JSON Objects

* Problem Statement:
  Create simple objects and array objects using JSON.

* Objective:
  To create simple JSON objects and JSON array objects.

* Outcome:
  Students will be able to:
  i) create simple JSON objects and JSON array object
  2) Insert them in a MongoDB database.

* S/W And H/W Requirements:
  MongoDB 4.4.1 server, IntelliJ IDE, Windows 10 i5 processor, etc.

Teacher's Sign.: ____________

Page No.
Date

* Theory:

- JSON with Java:
  Before we start with operations on JSON using Java, we need to install any of the JSON modules available.

- JSON
  - (an acronym for JavaScript Object Notation) is a lightweight data-interchange format and is most commonly used for client-server communication.
  - Its both easy to read/write and language independent.
  - A JSON value can be another JSON object, array, number, string, boolean or null

- JSON in Java
  - The 'JSON-Java' library is also known as org.json provides us with classes that are used to parse and manipulate JSON in Java
  - Furthermore, this library can also convert between JSON, XML, HTTP, Headers, Cookies, etc.

- JSON Object
  A JSONObject is an unordered collection of key & values pairs, resembling Java's native map implementations.

Teacher's Sign.: ____________

Page No.
Date

- keys are unique strings that cannot be null
- Values can be anything from a boolean, Number, String, JSON Array or even a JSON Object, NULL Object.

- A JSON Object can be represented by a string enclosed within curly braces with keys and values separated by a colon, and pairs seperated by comma.

- It has several constructors with which to construct a JSON Object.

- It also supports following main methods:
  (1) get (String key)
  - gets the object associated with the supplied key, throws JSON Exception if the key is not found.
  (2) opt (String key) - gets object associated with the supplied key, null otherwise.
  (3) put (String key, Object value).
  - inserts or replace a key-value pair is current JSON Object.

Syntax:

```
JSONObject jo = new JSONObject();
  jo.put ("name", "Varun");
  jo.put ("age", 20);
```

JSON Object : { "name":" Varun", "age": 20}

Teacher's Sign.:

Page No.
Date

- JSON Array:
  - its an ordered collection of values, resembling Java's native vector implementation.
  - values can be anything.
  - Its represented by a String wrapped within square Brackets and consists of a collection of values seperated by comma
  - Like JSON Object, it has a constructor that accepts source string and ~~process~~ parses it to construct a JSON Array.

Methods:
1. get (int index)
2. opt (int index)
3. put (Object value)

Syntax:
```
JSON Array ja = new JSONArray ( );
ja. put( Boolean TRUE):
ja put( "P") ;

JSONObject jo = new JSONObject ( );
    jo. put ("Array", ja );
```
* Creating JSONArray from an Array
```
• List <String> list = new ArrayList <> ( ):
  list. add ("A"):
  list. add ("B"):

  JSONArray ja = new JSONArray (list);
```

Teacher's Sign.: ____________

Page No.
Date

* Test Cases:

	Input	Expected Output	Result

i)	ID: 101
	Name: Varun
	Age: 20
	Movies
	Movie id: 101
	Genre: comedy
	Rating: 9.5
	Movie id = 102
	Genre: horror
	Rating: 8.9
		Document{{ id=101,
		Name = Varun,
		Age = 20,
		Movies = [Document{
		Movie id: 101,
		Genre: comedy,
		Rating = 9.5}},
		{ Movie id: 102,
		Genre: horror,
		Rating: 8.9}}] }}
			Success

* Conclusion:
Thus we created simple JSON objects and array objects in Java

Teacher's Sign.: ____________