########
JSON
########

-> JSON stands for Java Script object notation

-> JSON will represent data in key-value format

Ex :

```
{
     "id" : 101,
     "name: "raju",
     "age" : 20
}
```

-> JSON is intereoperable (language in-dependent & platform independent)

-> JSON is light weight

-> JSON is both human readable and machine readable format

-> In today's world people are using JSON format to exchange the data in B2B communications

-> Now a days JSON is having more demand than XML because of its simplicity and light weight

-> XML represents data in tags format (open tag & closed tag)
-> Meta data will be more than actual data in XML
-> XML occupies more memory to represent data

-> JSON will take less memory
-> JSON is light weight

-> To work with JSON data in Java Applications we have below 3rd party APIs

1) JACKSON API

2) GSON API

-> By using above apis we can convert JSON data to Java Object and vice versa

-> The process of converting Java Object into JSON is called as Serialization

-> The process of converting JSON data to Java Object is called as De-Serialization

1) Create Maven project with below dependencies

```
        <dependencies>
                <dependency>
```

```
                        <groupId>com.fasterxml.jackson.core</groupId>
                        <artifactId>jackson-databind</artifactId>
                        <version>2.13.3</version>
                </dependency>
                <dependency>
                        <groupId>org.projectlombok</groupId>
                        <artifactId>lombok</artifactId>
                        <version>1.18.24</version>
                        <scope>provided</scope>
                </dependency>
        </dependencies>
```

2) Create Java classes to represent data (Use lombok)

```
@Data
public class Author {

        private String authorName;
        private String authorEmail;
        private Long authorPhno;

}

@Data
public class Book {

        private Integer id;
        private String name;
        private Double price;
        private Author author;
}
```

3) Create Java class to convert Java Obj to JSON file

```
public class JavaToJsonConverter {

        public static void main(String[] args) throws Exception {

                Author author = new Author();
                author.setAuthorName("Rod Johnson");
                author.setAuthorEmail("r.john@gmail.com");
                author.setAuthorPhno(86868686l);

                Book book = new Book();
                book.setId(101);
                book.setName("Spring");
                book.setPrice(450.00);
                book.setAuthor(author);

                ObjectMapper mapper = new ObjectMapper();

                // converting java obj to json and store into a file
                mapper.writeValue(new File("book.json"), book);

                System.out.println("Conversion Completed....");
        }
}
```

4) Create Java Class To Convert JSON to Java Object

```
public class JsonToJavaConverter {

        public static void main(String[] args) throws Exception {

                File jsonFile = new File("book.json");

                ObjectMapper mapper = new ObjectMapper();

                Book book = mapper.readValue(jsonFile, Book.class);

                System.out.println(book);

        }
}
```

++++++++++++++++++++++
Working with GSON API
+++++++++++++++++++++++

1) Create a maven project with below dependency

```
<dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.9.0</version>
</dependency>
```

-> GSON api provided by google

-> In GSON api we have 'Gson' class to perform conversions

toJson ( ) -> to convert java object to JSON

fromJson ( ) -> to convert json data to java object