JavaScript Objects Practice Exercises

---

1 – Basic Objects

1. **Create a Person Object**
   - Create an object called `person` with properties: `firstName`, `lastName`, `age`, `city`.
   - Print each property individually using `console.log()`.
2. **Update Properties**
   - Create an object `book` with properties: `title`, `author`, `pages`.
   - Update the `pages` property to a new value and print the updated object.
3. **Add a New Property**
   - Create an object `movie` with properties: `name`, `year`, `rating`.
   - Add a new property `genre` and assign a value.

---

2 – Arrays and Nested Objects

4. **Array Inside Object**
   - Create an object `playlist` with properties: `name` and `songs` (array of 3 songs).
   - Print all songs.
   - Print the first and last song from the array.
5. **Nested Object**
   - Create an object `student` with properties: `name`, `age`, and `address` (nested object with `street`, `city`, `postalCode`).
   - Print the `city` and `postalCode` of the student.
6. **Multi-Level Nested Object**
   - Add a nested `grades` object inside `student` with subjects like `math` and `science`.
   - Print the math grade.

---

3 – Methods (Functions) Inside Objects

7. **Non-Parameterized Function**
   - Create an object `car` with properties like `brand` and a method `start()` that prints a message when the car starts.
   - Call the `start()` method.
8. **Parameterized Function**

- Add a method `service` to the `car` object which accepts a `technicianName` as a parameter and prints a message showing who serviced the car.
- Call the method with different names.

9. **Method Using `this`**
   - Add a method `showDetails` inside the `car` object that prints the car's brand and year using `this`.

---

4 – Real-World Object Practice

10. **Movie Booking Scenario**
    - Create an object `movieTicket` with properties: `movieName`, `theatre`, `seatsAvailable`, `isAvailable`, and a method `bookTicket(numSeats)`.
    - Inside the method, reduce `seatsAvailable` if enough seats exist; otherwise print "Not enough seats available."
    - Test booking tickets multiple times.
11. **Student Report**
    - Create a `student` object with properties: `name`, `age`, `subjects` (array), `grades` (nested object).
    - Add a method `calculateAverage()` to calculate and print the average of all grades.
    - Print student name and average grade.