Marvellous Study Tracker App Interview Questions

Q1: What is the purpose of your Study Tracker project?

A: The project is designed to help students maintain a record of their study sessions, view summaries, and export their logs for tracking productivity and planning better study schedules.

Q2: Why did you use `ArrayList` instead of an array?

A: I used `ArrayList` because the number of study logs is dynamic and can grow or shrink as needed. Unlike arrays, `ArrayList` provides built-in methods like `add()` and is more flexible for data storage.

Q3: What is the role of `TreeMap` in this project?

A: `TreeMap` is used in **SummaryByDate** and **SummaryBySubject** to store data in a **sorted order (ascending)** based on keys (date or subject). This makes the summary output structured and easy to read.

Q4: How is data persistence handled?

A: Currently, the project stores data in memory during execution and allows exporting data to a **CSV file** for future reference. Permanent database storage can be added later using MySQL or any database.

Q5: Why did you use `LocalDate` instead of manually entering the date?

A: I used `LocalDate.now()` to automatically capture the current date, ensuring that logs are timestamped accurately without requiring manual input.

Q6: What is the use of `try-with-resources` in ExportCSV()?

A: `try(FileWriter fwobj = new FileWriter(FileName))` ensures that the **file is automatically closed** after writing, even if an exception occurs, preventing resource leaks.

Q7: If you had to improve this project, what features would you add?

A:

- GUI interface using **Java Swing or JavaFX**.
- Option to **search logs by subject or date**.
- **Database integration** for permanent storage.
- **Graphical reports** (bar charts or pie charts) for visualization of study hours.
- Option to **edit or delete logs**.

Q8: What Java concepts are demonstrated in this project?

A:

- **OOP Concepts:** Classes, Objects, Constructors, Encapsulation.
- **Collections Framework:** ArrayList, TreeMap.
- **File Handling:** Writing to CSV file.
- **Exception Handling:** Try-catch blocks.
- **Date and Time API:** LocalDate class.
- **Loops and Conditional Statements.**

Q9: What is the time complexity of SummaryByDate()?

A:

- Each log is inserted into a TreeMap (O(log n)) where n is the number of logs.
- Total complexity = **O(n log n)** for building the summary.

Q10: What will happen if two study logs are added for the same date and subject?

A: The **SummaryByDate** will combine total hours for the date, and **SummaryBySubject** will combine hours for that subject. In the **DisplayLog**, both entries will be shown separately.