

Week 4 – Assignment

C#	<ol style="list-style-type: none">1. Build a Date-Time Calculator, with menu options; Inputs to be taken from the console and output to be displayed to the console;<ul style="list-style-type: none">- Add, Subtract between two dates and express the output in days, dates, weeks, months; (Involves String to Dates conversion);- Add, Subtract 'n' Days, Months, Weeks to the given date and derive the final date- Determine the Day of the Week for a given date- Determine the Week number for a given a date- Translate natural language phrases like "Today, Tomorrow, Day-after-tomorrow, yesterday, Day-before-yesterday, Last week, Previous week, Next week, Next month, Next Year, Last month, Last year, Month after, Month before, n weeks from now, n days from now, n months from now, n years from now, n days earlier, n weeks earlier, n months earlier, n years earlier etc."2. Extend the Date-Time Calculator to capture and display history of the calculations in the session in Memory3. Extend the Date-Time Calculator to support multiple languages (Ex: English, Hindi, French etc.)4. Extend the Date-Time Calculator to capture, persist (file storage) and display history for the last 100 sessions (the display needs to be paused & continued) - Using Serialization, Custom Data Structures & text, binary File formats, CSV, XML, YAML, JSON5. Extend the Date-Time Calculator to support bulk processing & operations - A file of 100,000 operations;6. Convert the OOPS based design to Functional programming using lambdas7. Automate the data-creation part of generating the file with 100,000 operations <p>* Apply/Practice TDD, BDD while doing the assignment; Outputs should include unit test report;</p>
----	---

	<p>>> Ensure the session storage captures info. Using different data types --> Strings, Dates, and Integers/Longs etc.</p> <p>>> Extensions and features to be built without changing existing code as much as possible</p> <p>>> Those who demonstrate Innovative solutions (ex: not using 'if/switch' conditions for functions), Engg. practices, Optimized code will be awarded brownie points</p>
Postgresql	<ul style="list-style-type: none"> - Implement the Date-Time Calculator as db operations, stored-procs & queries - Write queries to process the file with 100,000 operations and store the outputs back in a file - using DB - Build a dashboard from the Session data to indicate: <ul style="list-style-type: none"> - Overall Statistics across all sessions, per operation - Sessions wise statistics, per operation - Operations wise statistics per Session & across all sessions