Go Lang Live Coding Test

Instructions for Candidate:

- Please read each question carefully.
- Write your Go code solution in the shared document provided.
- The use of AI is not permitted
- Ensure your code is complete and runnable for each question, including `package main`, necessary `import` statements, and a `func main()` to demonstrate your solution where applicable.
- You have approximately 30 minutes to complete all questions.

Question 1: Structs and Methods (Approx. 7 minutes)

Define a struct in Go called `Product` with two fields:

1. `Name` (string)
2. `Price` (float64)

Then, write a method for this `Product` struct called `GetPriceWithTax`. This method should take a `taxRate` (float64, e.g., 0.1 for 10% tax) as an argument and return the price of the product *including* the tax.

Finally, show how you would create an instance of `Product` with Name "Laptop" and Price 1200.0, and then call `GetPriceWithTax` with a tax rate of 0.07 (7%) and print the result to the console.

Your Answer :

Question 2: Functions, Slices, and Error Handling (Approx. 8 minutes)

Write a Go function called `CalculateAverage` that:

1. Takes a slice of integers (`[]int`) as input.
2. Returns two values:
    - The average of the numbers in the slice as a `float64`.
    - An `error`.
3. If the input slice is empty, the function should return `0.0` for the average and an error message "cannot calculate average of an empty slice".
4. Otherwise, it should calculate and return the average and `nil` for the error.

Show how you would call this function within `func main()`:

- First, with a sample slice `[]int{10, 20, 30, 40}`. Print the average if there's no error, or print the error if one occurs.
- Second, with an empty slice. Print the error if one occurs.

Your Answer :

Question 3: Maps and Iteration (Approx. 8 minutes)

Declare a map in Go called `inventory` where keys are product names (strings) and values are their quantities (integers). Initialize this map with the following data:

- `"apple"`: 10
- `"banana"`: 5
- `"orange"`: 0

Write Go code (within `func main()`) to iterate through this `inventory` map. For each product:

- If the quantity is 0, print: `"[ProductName] is out of stock."` (e.g., "orange is out of stock.")
- Otherwise, print: `"[ProductName] has [Quantity] items."` (e.g., "apple has 10 items.")

Your Answer :