GO LANG

PAVITHARANI G P
2347244

1. Case Study: Temperature Converter

Problem Statement:

You are tasked with creating a temperature converter program in Go. The program should allow users to convert temperatures between Celsius, Fahrenheit, and Kelvin.

1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
3. Celsius to Kelvin
4. Kelvin to Celsius
5. Fahrenheit to Kelvin
6. Kelvin to Fahrenheit

CODE :

```go
//2347244

package main

import (
    "fmt"
)

func main() {
    var choice int
    var temperature float64

    fmt.Println("Temperature Converter")
    fmt.Println("1. Celsius to Fahrenheit")
    fmt.Println("2. Fahrenheit to Celsius")
    fmt.Println("3. Celsius to Kelvin")
    fmt.Println("4. Kelvin to Celsius")
    fmt.Println("5. Fahrenheit to Kelvin")
    fmt.Println("6. Kelvin to Fahrenheit")
    fmt.Print("Enter your choice: ")
    fmt.Scanln(&choice)

    fmt.Print("Enter temperature: ")
    fmt.Scanln(&temperature)

    switch choice {
```

```
    case 1:
        fmt.Printf("%.2f°C is %.2f°F\n", temperature,
celsiusToFahrenheit(temperature))
    case 2:
        fmt.Printf("%.2f°F is %.2f°C\n", temperature,
fahrenheitToCelsius(temperature))
    case 3:
        fmt.Printf("%.2f°C is %.2fK\n", temperature, celsiusToKelvin(temperature))
    case 4:
        fmt.Printf("%.2fK is %.2f°C\n", temperature, kelvinToCelsius(temperature))
    case 5:
        fmt.Printf("%.2f°F is %.2fK\n", temperature,
fahrenheitToKelvin(temperature))
    case 6:
        fmt.Printf("%.2fK is %.2f°F\n", temperature,
kelvinToFahrenheit(temperature))
    default:
        fmt.Println("Invalid choice!")
    }
}

func celsiusToFahrenheit(celsius float64) float64 {
    return (celsius * 9/5) + 32
}

func fahrenheitToCelsius(fahrenheit float64) float64 {
    return (fahrenheit - 32) * 5 / 9
}

func celsiusToKelvin(celsius float64) float64 {
    return celsius + 273.15
}

func kelvinToCelsius(kelvin float64) float64 {
    return kelvin - 273.15
}

func fahrenheitToKelvin(fahrenheit float64) float64 {
    return (fahrenheit - 32) * 5/9 + 273.15
}

func kelvinToFahrenheit(kelvin float64) float64 {
    return (kelvin - 273.15) * 9/5 + 32
}
```

OUTPUT :

```
Temperature Converter
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
3. Celsius to Kelvin
4. Kelvin to Celsius
5. Fahrenheit to Kelvin
6. Kelvin to Fahrenheit
Enter your choice: 1
Enter temperature: 22
22.00°C is 71.60°F
gppavitharani@Its-Paviii CASE STUDY % go run '/Users/gppavitharani/Documents/3TRIMESTER/G
OLANG/CASE STUDY/TEMP.go'
Temperature Converter
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
3. Celsius to Kelvin
4. Kelvin to Celsius
5. Fahrenheit to Kelvin
6. Kelvin to Fahrenheit
Enter your choice: 2
Enter temperature: 33
33.00°F is 0.56°C
gppavitharani@Its-Paviii CASE STUDY % go run '/Users/gppavitharani/Documents/3TRIMESTER/G
OLANG/CASE STUDY/TEMP.go'
Temperature Converter
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
3. Celsius to Kelvin
4. Kelvin to Celsius
5. Fahrenheit to Kelvin
6. Kelvin to Fahrenheit
Enter your choice: 3
Enter temperature: 28
28.00°C is 301.15K
gppavitharani@Its-Paviii CASE STUDY % go run '/Users/gppavitharani/Documents/3TRIMESTER/G
OLANG/CASE STUDY/TEMP.go'
Temperature Converter
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
3. Celsius to Kelvin
4. Kelvin to Celsius
5. Fahrenheit to Kelvin
6. Kelvin to Fahrenheit
Enter your choice: 4
Enter temperature: 305
305.00K is 31.85°C
gppavitharani@Its-Paviii CASE STUDY % go run '/Users/gppavitharani/Documents/3TRIMESTER/G
OLANG/CASE STUDY/TEMP.go'
Temperature Converter
1. Celsius to Fahrenheit
2. Fahrenheit to Celsius
3. Celsius to Kelvin
4. Kelvin to Celsius
5. Fahrenheit to Kelvin
6. Kelvin to Fahrenheit
Enter your choice: 5
Enter temperature: 34
34.00°F is 274.26K
```

Evaluated By	Comments For Documentation	Usage Of Different Datatypes	Usage of Maps	Usage of branching (if and switch)	Usage of loops	Completeness	Concept Clarity
Shah Riya Pranav	**Good documentation**	**Used int and float datatypes**	**Not necessary**	**Switch is used**	**Not needed**	**Completed**	**Knows the concept**

2. Use Switch case for branching
Create User defined functions for every conversion (6 User Defined Function)
Case Study: Scientific Calculator
Problem Statement:
Develop a Scientific calculator program in Go that performs

1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exponentiation (x^y)
6. Square root (√x)

Use Switch case for branching
Use Printf function to print with types

CODE :

```go
//2347244

package main

import (
   "fmt"
   "math"
)

func addition(x, y float64) float64 {
   return x + y
}

func subtraction(x, y float64) float64 {
   return x - y
}

func multiplication(x, y float64) float64 {
   return x * y
}

func division(x, y float64) float64 {
   if y == 0 {
       fmt.Println("Error: Division by zero")
       return 0
   }
   return x / y
```

```
}

func exponentiation(x, y float64) float64 {
    return math.Pow(x, y)
}

func squareRoot(x float64) float64 {
    if x < 0 {
        fmt.Println("Error: Cannot calculate square root of a negative number")
        return 0
    }
    return math.Sqrt(x)
}

func main() {
    var choice int
    var x, y float64

    fmt.Println("Scientific Calculator")
    fmt.Println("1. Addition (+)")
    fmt.Println("2. Subtraction (-)")
    fmt.Println("3. Multiplication (*)")
    fmt.Println("4. Division (/)")
    fmt.Println("5. Exponentiation (x^y)")
    fmt.Println("6. Square root (√x)")
    fmt.Print("Enter your choice: ")
    fmt.Scanln(&choice)

    switch choice {
    case 1:
        fmt.Print("Enter two numbers separated by space: ")
        fmt.Scanln(&x, &y)
        fmt.Printf("Result: %.2f\n", addition(x, y))
    case 2:
        fmt.Print("Enter two numbers separated by space: ")
        fmt.Scanln(&x, &y)
        fmt.Printf("Result: %.2f\n", subtraction(x, y))
    case 3:
        fmt.Print("Enter two numbers separated by space: ")
        fmt.Scanln(&x, &y)
        fmt.Printf("Result: %.2f\n", multiplication(x, y))
    case 4:
        fmt.Print("Enter two numbers separated by space: ")
        fmt.Scanln(&x, &y)
        fmt.Printf("Result: %.2f\n", division(x, y))
```

```
    case 5:
        fmt.Print("Enter base and exponent separated by space: ")
        fmt.Scanln(&x, &y)
        fmt.Printf("Result: %.2f\n", exponentiation(x, y))
    case 6:
        fmt.Print("Enter a number: ")
        fmt.Scanln(&x)
        fmt.Printf("Result: %.2f\n", squareRoot(x))
    default:
        fmt.Println("Invalid choice")
    }
```

OUTPUT :

```
Scientific Calculator
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exponentiation (x^y)
6. Square root (√x)
Enter your choice: 1
Enter two numbers separated by space: 66 4
Result: 70.00
gppavitharani@Its-Paviii CASE STUDY % go run '/Users/gppavitharani/Documents/3TRIMESTER/GOLANG/CASE STUDY/Calc.go'
Scientific Calculator
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exponentiation (x^y)
6. Square root (√x)
Enter your choice: 2
Enter two numbers separated by space: 88 22
Result: 66.00
gppavitharani@Its-Paviii CASE STUDY % go run '/Users/gppavitharani/Documents/3TRIMESTER/GOLANG/CASE STUDY/Calc.go'
Scientific Calculator
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exponentiation (x^y)
6. Square root (√x)
Enter your choice: 3
Enter two numbers separated by space: 54 8
Result: 432.00
gppavitharani@Its-Paviii CASE STUDY % go run '/Users/gppavitharani/Documents/3TRIMESTER/GOLANG/CASE STUDY/Calc.go'
Scientific Calculator
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exponentiation (x^y)
6. Square root (√x)
Enter your choice: 4
Enter two numbers separated by space: 87 3
Result: 29.00
gppavitharani@Its-Paviii CASE STUDY % go run '/Users/gppavitharani/Documents/3TRIMESTER/GOLANG/CASE STUDY/Calc.go'
Scientific Calculator
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exponentiation (x^y)
6. Square root (√x)
Enter your choice: 5
Enter base and exponent separated by space: 5 7
Result: 78125.00
```

Evaluated By	Comments For Documentation	Usage Of Different Datatypes	Usage of Maps	Usage of branching (if and switch)	Usage of loops	Completeness	Concept Clarity
Shah Riya Pranav	**Good**	**Used 2 datatypes**	**Not necessary**	**Switch is used**	**Not necessary**	**Completed**	**Knows the concept of switch case**

3. Case Study: Online Bookstore Inventory Checker

Problem Statement:

You're tasked with creating a program for an online bookstore to check the inventory of books.

The program should allow users to search for books by title and display whether the book is available in the inventory or not.

Create a struct named Book with title and availability (boolean)

Create a map named Inventory with Book title as key and availability (boolean) as value

Check the availability with if condition and print whether the book is available or not

```
//2347244

package main

import (
    "fmt"
)

type Book struct {
    Title string
    Available bool
}

func main() {
    // Initialize inventory with some books
    inventory := map[string]Book{
        "The Lord of the Rings": {Title: "The Lord of the Rings", Available: true},
        "The Hitchhiker's Guide to the Galaxy": {Title: "The Hitchhiker's Guide to
the Galaxy", Available: false},
```

```
        "Pride and Prejudice": {Title: "Pride and Prejudice", Available: true},
    }

    // Get book title from user
    var bookTitle string
    fmt.Println("Enter book title: ")
    fmt.Scanln(&bookTitle)

    // Check availability
    book, ok := inventory[bookTitle]
    if ok {
        if book.Available {
            fmt.Println(bookTitle, "is available!")
        } else {
            fmt.Println(bookTitle, "is currently unavailable.")
        }
    } else {
        fmt.Println("Book not found in inventory.")
    }
}
```

OUTPUT :

```
gppavitharani@Its-Paviii CASE STUDY % go run Book.go
Enter book title:
book1
book1 is available!
gppavitharani@Its-Paviii CASE STUDY % go run Book.go
Enter book title:
TheHitchhiker'sGuidetotheGalaxy
TheHitchhiker'sGuidetotheGalaxy is currently unavailable.
```

Evaluated By	Comments For Documentation	Usage Of Different Datatypes	Usage of Maps	Usage of branching (if and switch)	Usage of loops	Completeness	Concept Clarity
Shah Riya Pranav	**Well cleared**	**Used string & boolean datatypes**	**Map is done**	**Done**	**Not needed**	**Completed**	**Concepts are cleared**

4. Case Study: Student Grade Calculator

Problem Statement:
You're tasked with creating a program to calculate the final grades of students based on their scores in multiple subjects.
The program should allow users to input scores for each subject, calculate the average score, and assign grades based on predefined grading criteria.

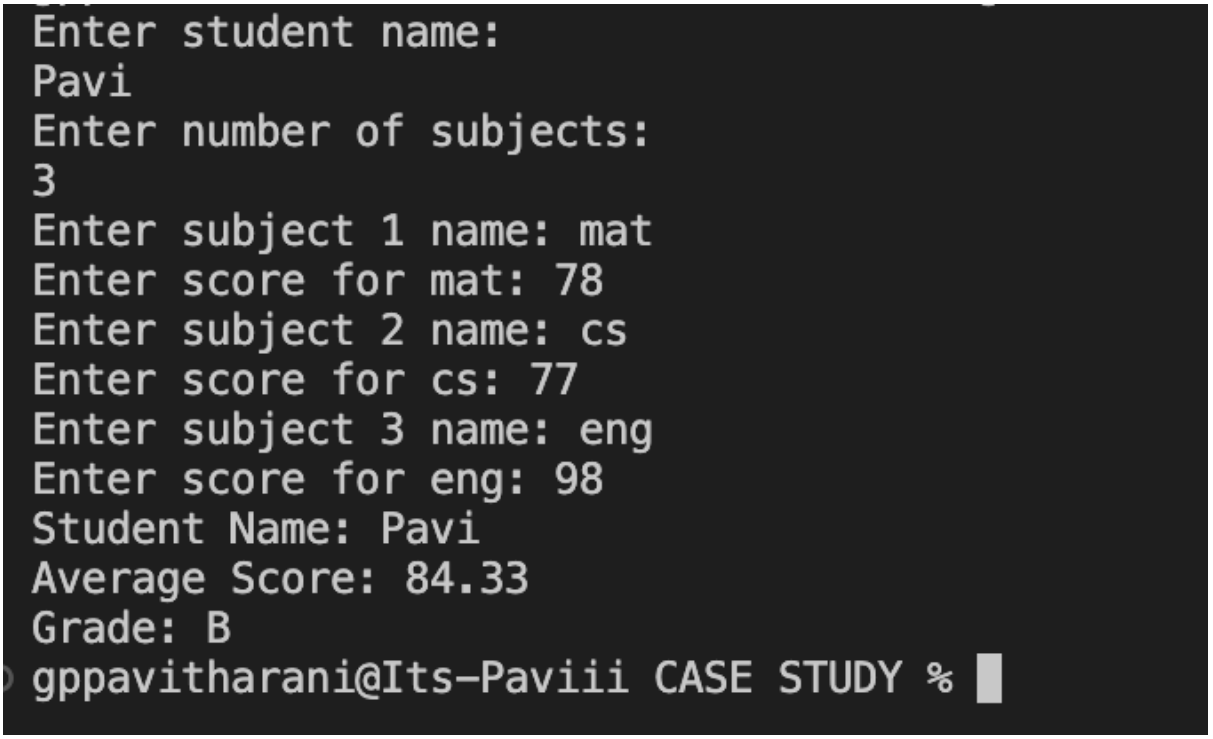
Use scanIn, Printf, loops, maps

CODE :

```go
package main

import (
    "fmt"
)

func main() {
    // Define grading criteria
    gradingMap := map[int]string{
        90: "A",
        80: "B",
        70: "C",
        60: "D",
        0:  "F",
    }

    // Initialize variables
    var numSubjects int
    var studentName string
    var subjectScores map[string]int

    // Get student name
    fmt.Println("Enter student name: ")
    fmt.Scanln(&studentName)

    // Get number of subjects
    fmt.Println("Enter number of subjects: ")
    fmt.Scanln(&numSubjects)

    // Create map to store subject scores
    subjectScores = make(map[string]int)

    // Get score for each subject
```

```
    for i := 1; i <= numSubjects; i++ {
        var subjectName string
        var score int

        fmt.Printf("Enter subject %d name: ", i)
        fmt.Scanln(&subjectName)

        fmt.Printf("Enter score for %s: ", subjectName)
        fmt.Scanln(&score)

        subjectScores[subjectName] = score
    }

    // Calculate total score and average
    var totalScore int
    for _, score := range subjectScores {
        totalScore += score
    }
    averageScore := float64(totalScore) / float64(numSubjects)

    // Determine grade based on grading criteria
    var grade string
    for threshold, letterGrade := range gradingMap {
        if averageScore >= float64(threshold) {
            grade = letterGrade
            break
        }
    }

    // Print results
    fmt.Printf("Student Name: %s\n", studentName)
    fmt.Printf("Average Score: %.2f\n", averageScore)
    fmt.Printf("Grade: %s\n", grade)
}
```

OUTPUT :



Evaluated By	Comments For Documentation	Usage Of Different Datatypes	Usage of branching (if and switch)	Usage of Maps	Usage of loops	Completeness	Concept Clarity
Shah Riya Pranav	**Well documented**	**Used**	**If is used**	**Done**	**Not necessary**	**Completed**	**Knows the concepts well and good**