Self Study Topics - Go Reference: https://golangr.com/download-answers/	
1	Variables
2	How to get Input
3	How to display an output
4	Basic data Types
5	Control Structures - if, switch - case, for, break, continue
6	Arrays, struct, Strings, Slices, Maps
7	Functions
Exercise: https://www.w3schools.com/go/exercise.php?filename=exercise_switch1	

Go Basics : Practice Programs (20-2-2025)

1. Create a Go program which prints “Hello World”

```
package main

import "fmt"

func main() {
fmt.Println("Bob")
}
```

2. Modify the below program from the console and check if it's between 50 and 100.

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

func main() {
    reader := bufio.NewReader(os.Stdin)

    fmt.Print("Enter a number: ")
    str1, _ := reader.ReadString('\n')

    // remove newline
    str1 = strings.Replace(str1,"\n","",-1)

    // convert string variable to int variable
    num, e := strconv.Atoi(str1)
    if e != nil {
    fmt.Println("conversion error:", str1)
    }

    if num >= 1 && num <= 10 {
    fmt.Println("correct")
    } else {
        fmt.Println("num not in range")
    }
}
```

3. Create a list of names using String Array and print them

```
package main

import (
    "fmt"
)

func main() {
    var a = []int64{ 1,2,3,4,5,6,7,8,9,10 }
    fmt.Println(a)
}
```

4. Modify the below program to define for loop to print odd numbers and even numbers continuously

```go
package main

import "fmt"

func main() {
   for x := 1; x <= 10; x++ {
        fmt.Printf("iteration x: %d\n", x)
   }
}
```

5. Write a list of cities to the file

```go
package main

import "os"

func main() {
    file, err := os.Create("file.txt")

    if err != nil {
        return
    }
    defer file.Close()

    [redacted]
    for i := 0; i < len(a); i++ {
        file.WriteString(a[i])
        file.WriteString("\n")
    }
}
```

6. Modify the below program to perform division using a function

```go
package main

import "fmt"

func main() {
   var a float64 = 3
   var b float64 = 9
   var ret = sum(a, b)
   fmt.Printf( "Value is : %.2f\n", ret )
}

func sum(num1, num2 float64) float64 {
   return num1+num2
}
```

7. Go Routine Program – Understand the working of this program and show its output

```go
package main

import (
    "fmt"
    "time"
)

func f(from string) {
    for i := 0; i < 3; i++ {
        fmt.Println(from, ":", i)
    }
}

func main() {

    f("direct")


    go f("goroutine")



    go func(msg string) {
        fmt.Println(msg)
    }("going")

    time.Sleep(time.Second)
    fmt.Println("done")
}
```

8. Study 3 RPC Programs uploaded. Check its output

9. Maps in GO

```go
import (
    "fmt"
    "maps"
)

func main() {

    m := make(map[string]int)


    m["k1"] = 7
    m["k2"] = 13

    fmt.Println("map:", m)


    v1 := m["k1"]
    fmt.Println("v1:", v1)

    v3 := m["k3"]
    fmt.Println("v3:", v3)

    fmt.Println("len:", len(m))


    delete(m, "k2")
    fmt.Println("map:", m)

    clear(m)
    fmt.Println("map:", m)

    _, prs := m["k2"]
    fmt.Println("prs:", prs)




    n := map[string]int{"foo": 1, "bar": 2}
    fmt.Println("map:", n)

    n2 := map[string]int{"foo": 1, "bar": 2}
    if maps.Equal(n, n2) {
        fmt.Println("n == n2")
    }
```

10. Implement an Acronym server using RPC.[Hint – use Map concept in server program to maintain a dictionary]
11.