NITTE EDUCATION TRUST

NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY
An Autonomous Institution Approved by UGC/AICTE/Govt. of Karnataka
Accredited by NBA (Tier-I) and NAAC 'A+' Grade
Affiliated to Visvesvaraya Technological University, Belagavi
Post Box No. 6429, Yelahanka, Bengaluru-560064, Karnataka, INDIA

KNOWLEDGE • CHARACTER • UNITY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

21ADA681 GO PROGRAMMING LABORATORY

NAME :_____

REGISTER NO. : _____

YEAR/SEM :_____

SECTION :_____

BRANCH : _____

RECORD NOTE BOOK

REGNO. . _____

Certified that this is a bonafide observation of Practical work done by Mr/Ms/Mrs……of the ……

Semester……. ..... Branch during the Academic year……in the… ..... laboratory.

Staff–in–Charge **Head of the Department**

Internal Examiner **External Examine**

LIST OF EXPERIMENTS - PART A

S.No	NAME OF THE PROGRAMS
1.	Write a Go program to print “Hello, world!” on the screen. Compile and run the program.
2.	Write a Go a program that shows your name and address.
3.	Write a Go a program adds a comment with your name.
4.	Write a Go a program to get a number from the console and check if it’s between 1 and 10.
5.	Write a Go a program which creates an array with the number 0 to 10.
6.	Write a Go a program that counts from 1 to 10.
7.	Write a program which prompts the user to enter a floating-point number and prints the integer which is a truncated version of the floating-point number that was entered. Truncation is the process of removing the digits to the right of the decimal place.
8.	Write a program which prompts the user to enter a string. The program searches through the entered string for the characters ‘I’, ‘a’, and ‘n’. The program should print “Found!” if the entered string starts with the character ‘I’, ends with the character ‘n’, and contains the character ‘a’. The program should print “Not Found!” otherwise. The program should not be case-sensitive, so it does not matter if the characters are upper-case or lower-case.

LIST OF EXPERIMENTS- PART B

S.No	NAME OF THE EXPERIMENT
1.	Write a Go a program to check if a file exists on your local disk or not.
2.	Write a Go a program to write a list of cities to a new file.
3.	Write a Go a program that takes the string ‘hello world’ and slice it in two .
4.	Write a Go a program which creates a method that sums two numbers.
5.	Write a Go a program which creates a method that calls another method.
6.	Write a Go program which prompts the user to enter integers and stores the integers in a sorted slice. The program should be written as a loop. Before entering the loop, the program should create an empty integer slice of size (length) 3. During each pass through the loop, the program prompts the user to enter an integer to be added to the slice. The program adds the integer to the slice, sorts the slice, and prints the contents of the slice in sorted order. The slice must grow in size to accommodate any number of integers which the user decides to enter. The program should only quit (exiting the loop) when the user enters the character ‘X’ instead of an integer.
7.	Write a Go program which prompts the user to first enter a name, and then enter an address. Your program should create a map and add the name and address to the map using the keys “name” and “address”, respectively. Your program should use Marshal() to create a JSON object from the map, and then your program should print the JSON object.
8.	Write a Go program which reads information from a file and represents it in a slice of structs.

PART A

PROGRAM -01

Write a Go program to print “Hello, world!” on the screen. Compile and run the program.

```
package main
import "fmt"
func main() {
	fmt.Println("Hello, World!")
}
```

Output :

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/___2go_build_awesomeProject
Hello, World!
```

PROGRAM -02

Write a Go a program that shows your name and address.

```
package main
import "fmt"
func main() {
	fmt.Println("Nitte Meenakshi Institute of Technology ")
	fmt.Println("P.B.No.6429. ")
	fmt.Println("Yelahanka, Bangalore 560064")
	fmt.Println(" Karnataka, India")
}
```

Output:

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/__go_build_awesomeProject1

Nitte Meenakshi Institute of Technology,
P.B.No.6429.
Yelahanka, Bangalore 560064,
Karnataka, India.
```

PROGRAM -03

Write a Go a program adds a comment with your name.

```
package main

import "fmt"

func main() {
	fmt.Println("NMIT")
	fmt.Println("NMIT is  my college")

}
```

Output:

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/__go_build_awesomeProject2
NMIT
NMIT is  my college
```

PROGRAM -04

Write a Go a program to get a number from the console and check if it's between 1 and 10.

```
package main

import (
	"fmt"
)

func main() {
	var number int

	fmt.Print("Enter a number: ")
	_, err := fmt.Scanf("%d", &number)
	if err != nil {
		fmt.Println("Invalid input. Please enter a valid number.")
		return
	}

	if number >= 1 && number <= 10 {
		fmt.Println("The number is between 1 and 10.")
	} else {
		fmt.Println("The number is not between 1 and 10.")
	}
}
```

Output:

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/__go_build_awesomeProject4
Enter a number: 1
The number is between 1 and 10.

Process finished with the exit code 0
```

PROGRAM -05

Write a Go a program which creates an array with the number 0 to 10.

```
package main

import (
	"fmt"
)

func main() {
	// Create an array with numbers from 0 to 10
	var numbers [11]int

	for i := 0; i <= 10; i++ {
		numbers[i] = i
	}

	// Print the array
	fmt.Println("Array of numbers from 0 to 10:")
	for _, num := range numbers {
		fmt.Println(num)
	}
}
```

Output:

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/__go_build_awesomeProject5
Array of numbers from 0 to 10:
0
1
2
3
4
5
6
7
8
9
10

Process finished with the exit code 0
```

PROGRAM -06

Write a Go a program that counts from 1 to 10.

```
package main

import "fmt"

func main() {
	var sum float32 = 0
	var i float32 = 1
	for i <= 10 {
		sum = sum + i
		i = i + 1
	}
	fmt.Println("sum of 1 to 10 numbers is ", sum)
}
```

Output:

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/___6go_build_awesomeProject
sum of 1 to 10 numbers is  55

Process finished with the exit code 0
```

PROGRAM -07

Write a program which prompts the user to enter a floating-point number and prints the integer which is a truncated version of the floating-point number that was entered. Truncation is the process of removing the digits to the right of the decimal place.

```
package main

import "fmt"

func main() {
	var f float64
	fmt.Println("Enter a float:")
	fmt.Scanln(&f)

	i := int(f)
	fmt.Println("Truncated integer:", i)
}
```

Output:

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/__1go_build_awesomeProject1
Enter a float:
3.56
Truncated integer: 3
```

PROGRAM -08

Write a program which prompts the user to enter a string. The program searches through the entered string for the characters 'I', 'a', and 'n'. The program should print "Found!" if the entered string starts with the character 'I', ends with the character 'n', and contains the character 'a'. The program should print "Not Found!" otherwise. The program should not be case-sensitive, so it does not matter if the characters are upper-case or lower-case.

```
package main

import (
	"fmt"
	"strings"
)

func main() {
	str := "I am a batman"
	if strings.HasPrefix(str, "I") && strings.HasSuffix(str, "n") && strings.Contains(str,
"a") {
		fmt.Println("String found", str)
	} else {
		fmt.Println("String not found", str)
	}
}
```

Output:

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/__1go_build_awesomeProject2
String found I am a batman

Process finished with the exit code 0
```

PART – B

Program 01

Write a Go a program to check if a file exists on your local disk or not.

```
package main

import (
	"fmt"
	"os"
)

func main() {
	// Replace "yourfile.txt" with the path to the file you want to check
	filePath := "/home/nmit/a.py"

	// Check if the file exists
	if _, err := os.Stat(filePath); os.IsNotExist(err) {
		fmt.Printf("File does not exist: %s\n", filePath)
	} else if err != nil {
		fmt.Printf("Error checking file: %s\n", err)
	} else {
		fmt.Printf("File exists: %s\n", filePath)
	}
}
```

Output

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/__1go_build_awesomeProject3
File does not exist: /home/nmit/a.py

Process finished with the exit code 0
```

Program 02

Write a Go a program to write a list of cities to a new file.

```
package main

import (
	"fmt"
	"os"
)

func main() {
	// Define list of new cities
	newCities := []string{
		"London",
		"Paris",
		"Rome",
		"Berlin",
		"Madrid",
	}

	// Create a new file to store the list of new cities
	file, err := os.Create("new_cities.txt")
	if err != nil {
		panic(err)
	}
	defer file.Close()

	// Write the list of new cities to the file
	for _, city := range newCities {
		fmt.Fprintf(file, "%s\n", city)
	}

	// Print a message to indicate that the list of new cities has been saved
	fmt.Println("List of new cities saved to new_cities.txt")
}
```

Output:

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/__1go_build_awesomeProject4
List of new cities saved to new_cities.txt

Process finished with the exit code 0
```

Program 03

Write a Go a program that takes the string ‘hello world’ and slice it in two .

```
package main

import (
	"fmt"
)

func main() {
	// Original string
	str := "hello world"

	// Determine the midpoint
	mid := len(str) / 2

	// Slice the string into two parts
	firstPart := str[:mid]
	secondPart := str[mid:]

	// Print the results
	fmt.Println("First part:", firstPart)
	fmt.Println("Second part:", secondPart)
}
```

Output:

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/__1go_build_awesomeProject5
First part: hello
Second part:  world

Process finished with the exit code 0
```

PROGRAM 04

Write a Go a program which creates a method that sums two numbers.

```
package main

import (
	"fmt"
)

// sum function that takes two integers and returns their sum
func sum(a int, b int) int {
	return a + b
}

func main() {
	var num1, num2 int

	// Prompt the user to enter two numbers
	fmt.Print("Enter the first number: ")
	fmt.Scanf("%d", &num1)
	fmt.Print("Enter the second number: ")
	fmt.Scanf("%d", &num2)

	// Call the sum function and print the result
	result := sum(num1, num2)
	fmt.Printf("The sum of %d and %d is: %d\n", num1, num2, result)
}
```

Output :

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/___5go_build_awesomeProject
Enter the first number: 2
Enter the second number: 3
The sum of 2 and 3 is: 5

Process finished with the exit code 0
```

PROGRAM 05

Write a Go a program which creates a method that calls another method.

```
package main


import (
	"fmt"
)


// add function that takes two integers and returns their sum
func add(a int, b int) int {
	return a + b
}


// calculateAndPrint function that calls the add method and prints the result
func calculateAndPrint(num1 int, num2 int) {
	result := add(num1, num2)
	fmt.Printf("The sum of %d and %d is: %d\n", num1, num2, result)
}


func main() {
	var num1, num2 int


	// Prompt the user to enter two numbers
	fmt.Print("Enter the first number: ")
	fmt.Scanf("%d", &num1)
	fmt.Print("Enter the second number: ")
	fmt.Scanf("%d", &num2)
```

```
    // Call the calculateAndPrint function
    calculateAndPrint(num1, num2)
}
```

Output

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/__2go_build_awesomeProject1
Enter the first number: 1
Enter the second number: 3
The sum of 1 and 3 is: 4

Process finished with the exit code 0
```

PROGRAM 06

Write a Go program which prompts the user to enter integers and stores the integers in a sorted slice. The program should be written as a loop. Before entering the loop, the program should create an empty integer slice of size (length) 3. During each pass through the loop, the program prompts the user to enter an integer to be added to the slice. The program adds the integer to the slice, sorts the slice, and prints the contents of the slice in sorted order. The slice must grow in size to accommodate any number of integers which the user decides to enter. The program should only quit (exiting the loop) when the user enters the character 'X' instead of an integer.

```
package main

import (
	"fmt"
	"sort"
)

func main() {
	// Create an empty integer slice
	numbers := []int{}

	for {
		var input string
		fmt.Print("Enter an integer (or 'X' to exit): ")
		fmt.Scan(&input)

		// Check if the user wants to exit
		if input == "X" || input == "x" {
			break
		}

		// Try to convert the input to an integer
		var number int
```

```
            _, err := fmt.Sscanf(input, "%d", &number)
            if err != nil {
                fmt.Println("Invalid input. Please enter an integer.")
                continue
            }

            // Add the number to the slice
            numbers = append(numbers, number)

            // Sort the slice
            sort.Ints(numbers)

            // Print the sorted slice
            fmt.Println("Sorted numbers:", numbers)
        }
}
```

Output:

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/__2go_build_awesomeProject2
Enter an integer (or 'X' to exit): 3
Sorted numbers: [3]
Enter an integer (or 'X' to exit):
```

PROGRAM 07

Write a Go program which prompts the user to first enter a name, and then enter an address. Your program should create a map and add the name and address to the map using the keys “name” and “address”, respectively. Your program should use Marshal() to create a JSON object from the map, and then your program should print the JSON object.

```
package main

import (
	"encoding/json"
	"fmt"
)

func main() {
	// Create a map to hold the name and address
	data := make(map[string]string)

	// Prompt the user for their name
	fmt.Print("Enter your name: ")
	var name string
	fmt.Scanln(&name)

	// Prompt the user for their address
	fmt.Print("Enter your address: ")
	var address string
	fmt.Scanln(&address)

	// Add name and address to the map
	data["name"] = name
```

```
	data["address"] = address

	// Marshal the map to JSON
	jsonData, err := json.Marshal(data)
	if err != nil {
		fmt.Println("Error marshaling to JSON:", err)
		return
	}

	// Print the JSON object
	fmt.Println("JSON object:", string(jsonData))
}
```

Output:

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/__2go_build_awesomeProject3
Enter your name: NMIT
Enter your address: YELAHANKA
JSON object: {"address":"YELAHANKA","name":"NMIT"}

Process finished with the exit code 0
```

Program -8

Write a Go program which reads information from a file and represents it in a slice of structs. Assume that there is a text file which contains a series of names. Each line of the text file has a first name and a last name, in that order, separated by a single space on the line. Your program will define a name struct which has two fields, fame for the first name, and lame for the last name. Each field will be a string of size 20 (characters).Your program should prompt the user for the name of the text file. Your program will successively read each line of the text file and create a struct which contains the first and last names found in the file. Each struct created will be added to a slice, and after all lines have been read from the file, your program will have a slice containing one struct for each line in the file. After reading all lines from the file, your program should iterate through your slice of structs and print the first and last names found in each struct.

Step 1 : Create a Text File: First, create a text file named `names.txt` (or any name you choose) and add a few lines with first and last names separated by a space. For example:

John Doe

Jane Smith

Alice Johnson

Step 2: Go Program: Here's the Go program that reads the names from the text file:

```
package main


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


// Define the Name struct
type Name struct {
        First string
        Last  string
```

```
}

func main() {
	var filename string

	// Prompt the user for the name of the text file
	fmt.Print("Enter the name of the text file: ")
	fmt.Scan(&filename)

	// Open the file
	file, err := os.Open(filename)
	if err != nil {
		fmt.Println("Error opening the file:", err)
		return
	}
	defer file.Close()

	// Create a slice to hold the names
	var names []Name

	// Read the file line by line
	scanner := bufio.NewScanner(file)
	for scanner.Scan() {
		line := scanner.Text()
		// Split the line into first and last name
		parts := strings.Fields(line)
		if len(parts) >= 2 {
			firstName := parts[0]
			lastName := parts[1]
			// Create a new Name struct and add it to the slice
```

```
                names = append(names, Name{First: firstName, Last: lastName})
            }
        }

        // Check for errors in reading the file
        if err := scanner.Err(); err != nil {
            fmt.Println("Error reading the file:", err)
            return
        }

        // Print the names from the slice
        fmt.Println("Names found in the file:")
        for _, name := range names {
            fmt.Printf("First Name: %s, Last Name: %s\n", name.First, name.Last)
        }
}
```

Output

```
/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/__2go_build_awesomeProject4
Enter the name of the text file: names.txt


/home/nmit/.cache/JetBrains/GoLand2024.2/tmp/GoLand/___5go_build_awesomeProject
Enter the first number: 2
Enter the second number: 3
The sum of 2 and 3 is: 5

Process finished with the exit code 0
```