**Department of Computer Science and Engineering**

**Amrita School of Engineering**

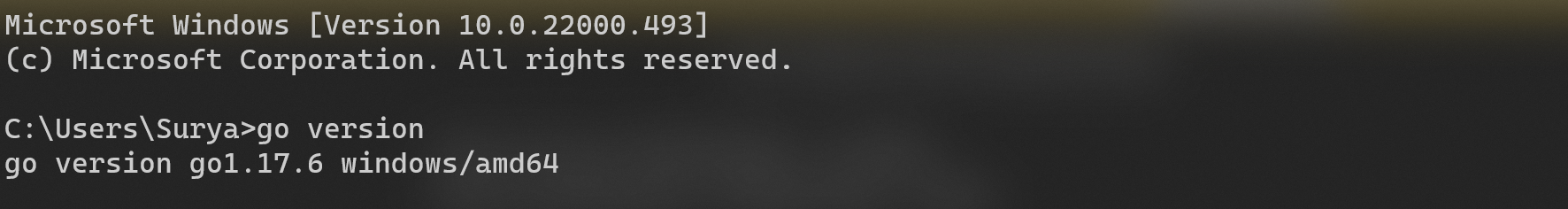
**Amrita Vishwa Vidyapeetham, Coimbatore**

**Lab : Basics of Go Programming**

**Suryanarayan.B**

**CB.EN.U4CSE19056**

1. GO environment



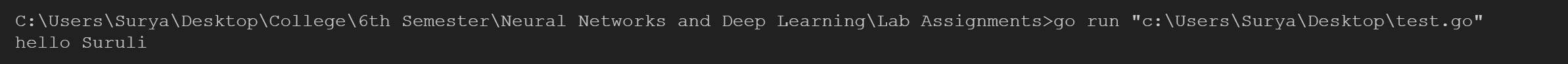
package main

import "fmt"

func main() {

    fmt.Println("hello Suruli")

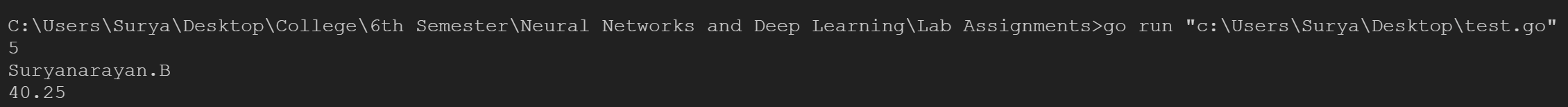
}



1. Datatypes supported in GO

| Data Type | Description |
| --- | --- |
| **int8** | 8-bit signed integer |
| **int16** | 16-bit signed integer |
| **int32** | 32-bit signed integer |
| **int64** | 64-bit signed integer |
| **uint8** | 8-bit unsigned integer |
| **uint16** | 16-bit unsigned integer |
| **uint32** | 32-bit unsigned integer |
| **uint64** | 64-bit unsigned integer |
| **int** | Both in and uint contain same size, either 32 or 64 bit. |
| **uint** | Both in and uint contain same size, either 32 or 64 bit. |
| **rune** | It is a synonym of int32 and also represent Unicode code points. |
| **byte** | It is a synonym of uint8. |
| **uintptr** | It is an unsigned integer type. Its width is not defined, but its can hold all the bits of a pointer value. |

1. Defining and declaring variables with different data types.
2. package main
3. import "fmt"
4. func main() {
5. var a int = 5
6. var b string = "Suryanarayan.B"
7. var fnum float64 = 40.25
8. fmt.Println(a)
9. fmt.Println(b)
10. fmt.Println(fnum)
11. }



4.Defining, writing and calling functions.

package main

import "fmt"

func addi(a, b int) int {

    Sum := a + b

    return Sum

}

func main() {

    fmt.Printf("Sum of the numbers : %d", addi(20, 15))

}



5.Different general purpose packages and their necessity for different functions.

1. Import paths: In Go language, every package is defined by a unique string and this string is known as import path. With the help of an import path, you can import packages in your program
   1. Example: import “fmt”
2. Package Declaration: In Go language, package declaration is always present at the beginning of the source file and the purpose of this declaration is to determine the default identifier for that package when it is imported by another package. For example: package main
3. Import declaration: The import declaration immediately comes after the package declaration. The Go source file contains zero or more import declaration and each import declaration specifies the path of one or more packages in the parentheses. For example:

// Importing single package

import "fmt"

// Importing multiple packages

import(

"fmt"

"strings"

"bytes"

)

1. Blank import: In Go programming, sometimes we import some packages in our program, but we do not use them in our program. When you run such types of programs that contain unused packages, then the compiler will give an error. So, to avoid this error, we use a blank identifier before the name of the package. For example:

import \_ "strings"

1. Nested Packages: In Go language, you are allowed to create a package inside another package simply by creating a subdirectory. And the nested package can import just like the root package. For example:

import "math/cmplx"

6.Defining and using arrays.

package main

import "fmt"

func main() {

    arr := [5]int{69, 420, 80, 11, 68}

    arrays(arr)

}

func arrays(arr [5]int) {

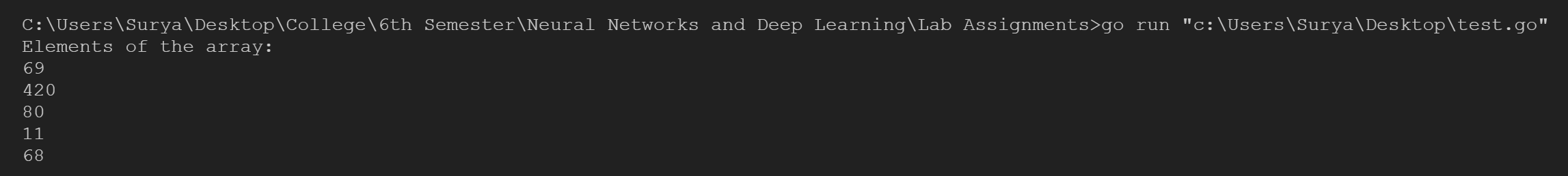
    fmt.Println("Elements of the array:")

    for i := 0; i < 5; i++ {

        fmt.Printf("%d\n", arr[i])

    }

}



7.Writing different conditional statements

package main

import "fmt"

func main() {

    var a int

    fmt.Scanln(&a)

    var b int

    fmt.Scanln(&b)

    if a > b {

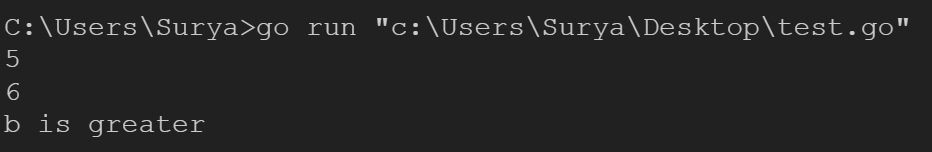
        fmt.Println("a is greater")

    } else {

        fmt.Println("b is greater")

    }

}



8.Writing different Loop structures for accessing arrays/ lists/ strings etc.

package main

import (

    "container/list"

    "fmt"

)

func main() {

    arr := [5]float64{45.2, 29.3, 39.5, 69.420, 420.69}

    var str string = "Hai"

    ls := list.New()

    ls.PushBack(10)

    ls.PushBack(12)

    ls.PushBack("Hai")

    ls.PushBack(45.69)

    strs(str)

    arrays(arr)

    fmt.Println("Elements of the list:")

    for e := ls.Front(); e != nil; e = e.Next() {

        fmt.Println(e)

    }

}

func arrays(arr [5]float64) {

    fmt.Println("Elements of the array:")

    for i := 0; i < 5; i++ {

        fmt.Printf("%f\n", arr[i])

    }

}

func strs(str string) {

    for i := 0; i < len(str); i++ {

        fmt.Printf("Character at %d Index Position = %c\n", i, str[i])

    }

}

