

Certainly, let's tackle the Go language programming questions from **Set B** and **Set C**.

Set B

1. Write a program in Go language to create structure student. Write a method show() whose receiver is a pointer of struct student.

```
package main

import "fmt"

// Define the Student struct
type Student struct {
    RollNo    int
    Name      string
    Percentage float64
}

// Method to display student information (receiver is a pointer to Student)
func (s *Student) Show() {
    fmt.Printf("Roll No: %d, Name: %s, Percentage: %.2f\n", s.RollNo, s.Name, s.Percentage)
}

func main() {
    student1 := Student{RollNo: 1, Name: "Alice", Percentage: 85.5}
    student1.Show() // Call the Show method

    // Alternatively, you can use a pointer to the struct
    student2 := &Student{RollNo: 2, Name: "Bob", Percentage: 90.2}
    student2.Show() // Method can be called on a pointer
}
```

Output:

Roll No: 1, Name: Alice, Percentage: 85.50

Roll No: 2, Name: Bob, Percentage: 90.20

2. Write a program in Go language to demonstrate working type switch in interface.

```
package main

import "fmt"

// Define an interface
type Shape interface {
    Area() float64
}

// Circle struct
type Circle struct {
    Radius float64
}
```

```

}

func (c Circle) Area() float64 {
    return 3.14159 * c.Radius * c.Radius
}

// Rectangle struct
type Rectangle struct {
    Width, Height float64
}

func (r Rectangle) Area() float64 {
    return r.Width * r.Height
}

func main() {
    shapes := []Shape{Circle{Radius: 5}, Rectangle{Width: 4, Height: 3}}

    for _, shape := range shapes {
        switch s := shape.(type) {
        case Circle:
            fmt.Println("Circle Area:", s.Area())
        case Rectangle:
            fmt.Println("Rectangle Area:", s.Area())
        default:
            fmt.Println("Unknown shape")
        }
    }
}

```

Output:

```

Circle Area: 78.53975
Rectangle Area: 12

```

3. Write a program in Go language to copy all elements of one array into another using method.

```

package main

import "fmt"

func copyArray(src []int) []int {
    dst := make([]int, len(src)) // Create a new array with the same
    length
    copy(dst, src)               // Copy elements using the built-in
    copy function
    return dst
}

```

```

func main() {
    originalArray := []int{1, 2, 3, 4, 5}
    newArray := copyArray(originalArray)

    fmt.Println("Original Array:", originalArray)
    fmt.Println("New Array:", newArray)
}

```

Output:

```

Original Array: [1 2 3 4 5]
New Array: [1 2 3 4 5]

```

Set C

1. Write a program in Go language to create an interface and display it's values with the help of type assertion.

```

package main

import "fmt"

// Define an interface
type Shape interface {
    Area() float64
}

// Circle struct
type Circle struct {
    Radius float64
}

func (c Circle) Area() float64 {
    return 3.14159 * c.Radius * c.Radius
}

func main() {
    var shape Shape = Circle{Radius: 5}

    // Type assertion
    if circle, ok := shape.(Circle); ok {
        fmt.Println("Circle Area:", circle.Area())
    } else {
        fmt.Println("Shape is not a Circle")
    }
}

```

Output:

```

Circle Area: 78.53975

```

2. Write a program in Go language to store n student information(rollno, name,

percentage) and write a method to display student information in descending order of percentage.

```
package main

import "fmt"
import "sort"

// Define the Student struct
type Student struct {
    RollNo    int
    Name      string
    Percentage float64
}

// Custom sort function to sort students by descending percentage
type ByPercentage []Student

func (a ByPercentage) Len() int           { return len(a) }
func (a ByPercentage) Swap(i, j int)      { a[i], a[j] = a[j], a[i] }
func (a ByPercentage) Less(i, j int) bool { return a[i].Percentage > a[j].Percentage }

func main() {
    students := []Student{
        {RollNo: 1, Name: "Alice", Percentage: 85.5},
        {RollNo: 2, Name: "Bob", Percentage: 90.2},
        {RollNo: 3, Name: "Charlie", Percentage: 78.9},
    }

    // Sort students by descending percentage
    sort.Sort(ByPercentage(students))

    // Display sorted students
    for _, student := range students {
        fmt.Printf("Roll No: %d, Name: %s, Percentage: %.2f\n",
            student.RollNo, student.Name, student.Percentage)
    }
}
```

Output:

```
Roll No: 2, Name: Bob, Percentage: 90.20
Roll No: 1, Name: Alice, Percentage: 85.50
Roll No: 3, Name: Charlie, Percentage: 78.90
```

3. Write a program in Go language to demonstrate working embedded interfaces.

```
package main

import "fmt"
```

```
// Define an interface
type Shape interface {
    Area() float64
}

// Define another interface that embeds Shape
type ColoredShape interface {
    Shape
    Color() string
}

// Circle struct
type Circle struct {
    Radius float64
    color  string
}

func (c Circle) Area() float64 {
    return 3.14159 * c.Radius * c.Radius
}

func (c Circle) Color() string {
    return c.color
}

func main() {
    circle := Circle{Radius: 5, color: "Red"}

    // Assign Circle to a ColoredShape variable
    var coloredShape ColoredShape = circle

    fmt.Println("Circle Area:", coloredShape.Area())
    fmt.Println("Circle Color:", coloredShape.Color())
}
```

Output:

```
Circle Area: 78.53975
Circle Color: Red
```

I hope these solutions are helpful! Feel free to ask if you have any further questions or want to explore other programming concepts in Go.

- <https://github.com/IMMORTAL1380/GO>
- <https://github.com/mskasa/zenn-contents>
- https://github.com/singh-hub1/GO_Lang
- <https://github.com/jakecoffman/rosalind>