**Structor**

package main

import (

    "fmt"

    "time"

)

type book struct {

    author      string

    name        string

    revision    int

    yearrelease time.Time

}

func main() {

    bookdetail := book{

        author:   "yagnik",

        name:     "Golang",

        revision: 3,

    }

    fmt.Println(bookdetail.author)

    fmt.Println(bookdetail.name)

    fmt.Println(bookdetail.revision, bookdetail.yearrelease)

}

yagnik

Golang

3 0001-01-01 00:00:00 +0000 UTC

**Structor as a function**

You can pass a structure as a function argument in very similar way as you pass any other variable or pointer. You would access structure variables in the same way as you did in the above example −

package main

import "fmt"

type book struct {

    author    string

    publisher string

    revision  int

}

func main() {

    var book1 book

    var book2 book

    book1.author = "yagnik"

    book1.publisher = "milman"

    book1.revision = 1

    book2.author = "mahamad"

    book2.publisher = "billigine"

    book2.revision = 2

    print\_func(book1)

    print\_func(book2)

}

func print\_func(book\_detail book) {

    fmt.Printf("%s \n", book\_detail.author)

    fmt.Printf("%s \n", book\_detail.publisher)

    fmt.Printf("%d \n", book\_detail.revision)

}

yagnik

milman

1

mahamad

billigine

2

**Structor as a pointer function**

**You can define pointers to structures in the same way as you define pointer to any other variable as follows −**

var struct\_pointer \*Books

struct\_pointer = &Book1;

struct\_pointer.title;

package main

import "fmt"

type book struct {

    author    string

    publisher string

    revision  int

}

func main() {

    var book1 book

    var book2 book

    book1.author = "yagnik"

    book1.publisher = "milman"

    book1.revision = 1

    book2.author = "mahamad"

    book2.publisher = "billigine"

    book2.revision = 2

    print\_func(&book1)

    print\_func(&book2)

}

func print\_func(book\_detail \*book) {

    fmt.Printf("%s \n", book\_detail.author)

    fmt.Printf("%s \n", book\_detail.publisher)

    fmt.Printf("%d \n", book\_detail.revision)

}

yagnik

milman

1

mahamad

billigine

2