**Declaring struct:**

type student struct{

name string

rollno int

city string

}

type student struct{

name, city string

rollno int

}

**Anonymous struct:**

var student struct{

name, city string

rollno int

}

func main() {

type student struct{

name, city string

rollno int

}

var student\_var student

fmt.Println(student\_var) // { 0}

var student1 struct{

name, city string

rollno int

}

fmt.Println(student1) // { 0}

}

**Initialize struct:**

func main() {

type student struct{

name, city string

rollno int

}

var student\_var student

student\_var = student{"rahul", "jaipur", 67}

fmt.Println(student\_var) // {rahul jaipur 67}

}

func main() {

type student struct{

name, city string

rollno int

}

var student\_var student

student\_var = student{rollno:67, city:"jaipur"} // { jaipur 67}

fmt.Println(student\_var)

// Anonymous struct

student1 := struct{

name string

city string

rollno int

}{

name : "rahul",

rollno :67,

}

fmt.Println(student1) // {rahul 67}

}

**Accessing struct’s field:**

func main() {

var student struct{

name, city string

rollno int

}

student.city = "jaipur"

student.rollno = 67

fmt.Println(student) // { jaipur 67}

type student1 struct{

name, city string

rollno int

}

var student\_var student1

student\_var.name = "rahul"

student\_var.rollno = 67

fmt.Println(student\_var) // {rahul 67}

}

**Pointer to struct:**

func main() {

type student struct{

name, city string

rollno int

}

student\_var := student{"rahul", "jaipur", 67}

p1 := &student\_var

fmt.Println(p1) // &{rahul jaipur 67}

fmt.Println(\*p1) // {rahul jaipur 67}

fmt.Println((\*p1).city) //jaipur

fmt.Println(p1.city) //jaipur

//Directly create a pointer

p2 := &student{"rahul", "jaipur", 67}

fmt.Println(p2) // &{rahul jaipur 67}

fmt.Println(\*p2) // {rahul jaipur 67}

fmt.Println((\*p2).city) //jaipur

fmt.Println(p2.city) //jaipur

//Pointer with anonymous struct

var student1 struct{

name, city string

rollno int

}

annony\_point := &student1

annony\_point.name = "rahul"

annony\_point.rollno = 67

fmt.Println(annony\_point) // &{rahul 67}

fmt.Println(\*annony\_point) // {rahul 67}

}

**Struct with new function:**

func main() {

type student struct{

name, city string

rollno int

}

p1 := new(student)

fmt.Println(\*p1) // { 0}

p1.city = "jaipur"

p1.rollno = 67

fmt.Println(\*p1) // { jaipur 67}

//anonymous struct with new

annony\_point := new(struct{

name, city string

rollno int

})

fmt.Println(\*annony\_point) // { 0}

annony\_point.name = "rahul"

fmt.Println(\*annony\_point) // {rahul 0}

}

**Anonymous struct fields:**

func main() {

type student struct{

string

int

}

a := student{"rahul", 67}

fmt.Println(a) // {rahul 67}

}

type student struct{

string

string

int

}

a := student{"rahul", "jaipur", 67}

fmt.Println(a) //error

func main() {

type student struct{

string

city string

int

}

a := student{"rahul","jaipur", 67}

fmt.Println(a) // {rahul jaipur 67}

}

**Nested struct:**

func main() {

type result struct{

math int

science int

english int

isPass bool

}

type student struct{

name, city string

rollno int

result result //Nested Struct as a field

}

a := student{

name: "Rahul",

city: "Jaipur",

rollno: 11,

result: result{

math: 45,

science: 78,

english: 90,

isPass: true,

},

}

fmt.Println("Final Result is: ", a.result.isPass) //true

}

**Embedded Struct:**

func main() {

type result struct{

math int

science int

english int

isPass bool

}

type student struct{

name, city string

rollno int

result

}

a := student{"rahul", "jaipur", 11, result{45, 78, 90, true}}

fmt.Println(a) // {rahul jaipur 11 {45 78 90 true}}

//anonymous nested embedded struct

result\_annon := struct{

math int

science int

english int

isPass bool

}{

math: 45,

science: 78,

english: 90,

isPass: true,

}

student\_annon := struct{

name, city string

rollno int

result

}{

name : "rahul",

city : "city",

rollno : 11,

result : result\_annon,

}

fmt.Println(student\_annon) // {rahul city 11 {45 78 90 true}}

}

**Passing struct to functions:**

package main

import "fmt"

func getValue(student\_copy1 \*student) string{

fmt.Println(student\_copy1) // &{rahul jaipur 67}

return student\_copy1.name

}

type student struct{

name, city string

rollno int

}

func main() {

a := student{"rahul", "jaipur", 67}

student\_copy := a

fmt.Println(student\_copy.rollno) // 67

fmt.Println(student\_copy) // {rahul jaipur 67}

// Passing address of struct to function

fmt.Println(getValue(&student\_copy)) // rahul

}

**Compare struct:**

func main() {

student := struct{

name, city string

rollno int

}{

name : "rahul",

city : "jaipur",

rollno : 11,

}

student1 := struct{

name, city string

rollno int

}{

name : "rahul",

city : "jaipur",

rollno : 11,

}

fmt.Println(student==student1) // true

student3 := struct{

name, city string

rollno int

}{

name : "rahul",

city : "jaipur",

}

fmt.Println(student==student3) // false

student4 := struct{

name, city string

rollno int

}{

name : "rahul",

city : "jaipur",

rollno : 0,

}

fmt.Println(student3==student4) // true

}