

Module 2 Quiz

Quiz, 7 questions

6/7 points (85.71%)



Congratulations! You passed!

Next Item



1 / 1
point

1.

Is the highlighted assignment to f in the following code a legal variable assignment?

```
var f func(string) int

func test(x string) int {
    return len(x)
}

func main() {
    f = test
}
```



Yes



Correct

Correct!



No

Module 2 Quiz

Quiz, 7 questions

0 / 1
point

6/7 points (85.71%)

2.

Which of the following statements correctly declares a function whose argument is another function which takes an integer as an argument and returns a string?

☒ func fA(fB (int) string)

This should not be selected

No func keyword used in the argument.

☐ func fA(fB func (int) string)

☐ func fA(fB func (int)) string

☐ func fA(fB func (string) int)

1 / 1
point

3.

What is an anonymous function?

☐ A function with no return value

☐ A function with multiple names

☒ A function with no name

Correct

Correct!

☐ A function with no arguments

Module 2 Quiz

Quiz, 7 questions

1 / 1
point**6/7 points (85.71%)**

4.

Which of the following statements correctly declares a function whose return value is another function which takes a string as an argument and returns an integer?

☐ `func fA(fB (int) string) func (string) int`☒ `func fA(fB func (int) string) {}`**Correct**

Correct!

☐ `func fA(int) string {}`☐ `func fA() fB func (string) int {}`

Module 2 Quiz

Quiz, 7 questions

1 / 1
point

6/7 points (85.71%)

5.

```
1 func fA() func() int {  
2     i := 0  
3     return func() int {  
4         i++  
5         return i  
6     }  
7 }  
8 func main() {  
9     fB := fA()  
10    fmt.Print(fB())  
11    fmt.Print(fB())  
12 }  
13
```

What does the above code print on the screen?

☒ 12**Correct**

Correct!

☐ 11☐ 01☐ 11 / 1
point

6.

What symbols are used in a function declaration to indicate that it is a variadic function?

☐ "->"☒ "..."**Correct**

Correct!

☐ "___"☐ "[]"

Module 2 Quiz

Quiz, 7 questions

1 / 1
point

6/7 points (85.71%)

7.

What does this routine produce?

```
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     i := 1
8
9     fmt.Print(i)
10
11    i++
12
13    defer fmt.Print(I+1)
14
15    fmt.Print(i)
16
17 }
```

☐ 132☐ 134☐ 234☒ 123**Correct**

Correct! The *defer* statement prints 3 after the surrounding function returns. Thus 2 is printed before 3.

