

Congratulations! You passed!

Grade received 85.71%

Latest Submission Grade 85.71%

To pass 75% or higher

Go to next
item

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

1 / 1 point

```
var f func(string) int

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

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

☒ Yes

☐ No

☒ **Correct**
Correct!

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?

1 / 1 point

☐ func fA(fB (int) string)

☒ func fA(fB func (int) string)

- ☐ func fA(fB func (int))
string
- ☐ func fA(fB func (string) int)
- ☒ **Correct**
Correct!

3. What is an anonymous function?

1 / 1 point

- ☐ A function with no return value
- ☐ A function with multiple names
- ☒ A function with no name
- ☐ A function with no arguments
- ☒ **Correct**
Correct!

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?

0 / 1 point

- ☐ func fA(fB (int) string) func (string) int
- ☐ func fA(fB func (int) string) {}
- ☐ func fA(int) string {}
- ☒ func fA() fB func (string) int{}

☒ **Incorrect**
The return value should
not have a name.

5.

```
1 func fA() func() int {  
2     i := 0  
3     return func() int {  
4         i++  
5         return i  
6     }  
7 }
```

1 / 1 p

```
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

☐ 11

☐ 01

☐ 1

☒ **Correct**
Correct!

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

1 / 1 point

☐ "->"

☒ "..."

☐ "___"

☐ "[]"

☒ **Correct**
Correct!

7. What does this routine produce?

1 / 1 p

```
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.