

The Go programming language has a pointer type. Why are pointers required in Go? What actions (or what data structures) would be impossible to make without pointers. Explain.

true or false: A language that has both statements and expressions cannot achieve perfect orthogonality. Explain (with examples)

Go defines a method to be a function that is bound to a structure. For instance

```
type deepak struct {
    chris int
    aline string
}
func (dianna deepak) toString() string {
    return fmt.Sprintf("%v %v", dianna.chris, dianna.aline)
}
```

Describe the differences between methods in Go and methods in Java/Kotlin.

Why is **circular garbage** a problem for **reference count** garbage collection and not for a problem for **stop-and-copy**? A complete answer will define the three terms in bold before addressing the “why” question.

A classic way of implementing enums is to assign each value to a single bit. For instance, if Kotlin used this strategy then the enum

```
class enum aa { A, B, C, D, E, F, G, H}
```

might use the following encoding

```
A 10000000
B 01000000
C 00100000
D 00010000
E 00001000
F 00000100
G 00000010
H 00000001
```

Discuss the plusses and minuses of this encoding approach for enums

The book comments that equality testing is more difficult than you might expect. Describe at least 3 issues that a general equality testing solution must address. Give examples in Kotlin that illustrate these issues.

Most statically typed languages developed since the 1970's use some form of name equivalence for types. Is static equivalence a bad idea? Why or why not?