Paradigm of Go

Go is an imperative language. It has terminating statements, including “return” and “goto” statement, “if” statement, “for” statement, “switch” statement, “select” statement, and so on. It also has statement for looping - “for” statement.

Citation: <https://golang.org/ref/spec#Statements>

Citation: <https://golang.org/ref/spec#For_statements>

Go is not exactly an object-oriented language. Although Go has types and methods and allows an object-oriented style of programming, there is no type hierarchy. Go also provides the implementation of interface. Go allows user to define “struct” type as an object, which is very similar to struct in C. However, unlike in C, Go also allows users to write method for struct type. In Go, there are also access control at package level defined by whether the method or attributes in a package is capitalized or not. Due to the lack of “class” features, access control is unlike many other object-oriented language, where access control is at class level.

Citation: <https://golang.org/doc/faq#Is_Go_an_object-oriented_language>