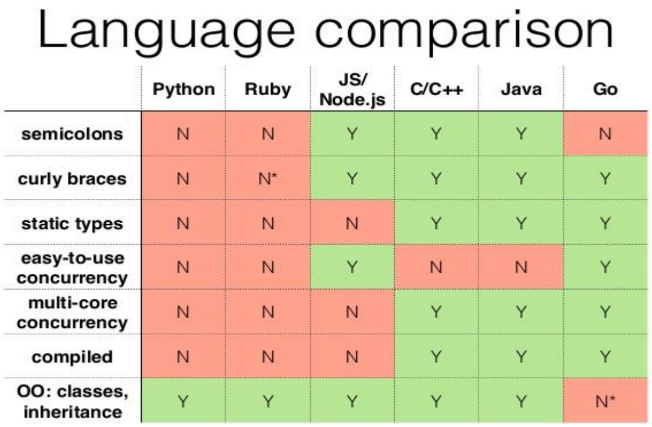
In past few days I was engaged in learning a new programming language called “Golang”, it’s a simple programming language for developing robust scalable software applications. I have explored many new things in the session, so I would like to share my thoughts about Golang programming language and my learning experience.

Go Vs other programming languages?



Language comparison

	Python	Ruby	JS/ Node.js	C/C++	Java	Go
semicolons	N	N	Y	Y	Y	N
curly braces	N	N*	Y	Y	Y	Y
static types	N	N	N	Y	Y	Y
easy-to-use concurrency	N	N	Y	N	N	Y
multi-core concurrency	N	N	N	Y	Y	Y
compiled	N	N	N	Y	Y	Y
OO: classes, inheritance	Y	Y	Y	Y	Y	N*

Golang is a general purpose programming language which can compete with C and C++ languages, because it has a very simple syntax, that helps us to achieve more things, instead of focusing our time with minor things. We can put our effort in higher level abstract things. When we look the syntax of Golang, it eliminates lot of things from C and other languages, example don’t need to specify a semicolon (;) to terminate the line, no access modifiers in Golang. We can do it by declaring the function name starting with upper case or lower case. Most important things are, its compilation time is much faster than other languages, and even compiling very large projects is done in few minutes.

Generally, it eliminates some standard patterns from Programming Language Theory, the language designers have made it simpler to achieve more performance in critical real world application. The compiler directly convert from high level language to machine language, i.e. we don’t need any intermediate steps like JVM for Java, so the compiler quickly build the application, so if we know the basic syntax in "Golang" we can develop our application with its inbuilt standard packages without depending on any 3rd party Frameworks/tools.

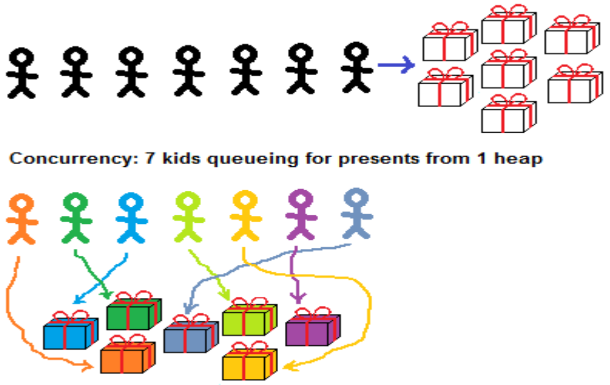
Go for Fresher's than experienced developers

I would say compare to the experienced developers, Golang will be more attractive to the fresh software engineers, because when we start working with Golang its provide very simple syntax to achieve many things. The learning curve is bit faster, but for the experience developers it will take some time to understand the core benefits of Golang programming language because for the past few years If someone developed their real world application using Microsoft technologies or any other technologies, it will not be so easy to quickly adapt this new language, it will take some time to digest or understand those things. Once the understanding comes one can build real world application and use it in a critical system development.

"Unlearn something to learn new things"

Go syntax is really different than other programming language, for example: We don't need while loop but we can achieve the same thing via "for" loop, so by learning one looping statement we can do the same thing what we did for other programming languages, so it gives flexibility to build the application.

Concurrency with GO



Concurrency: 7 kids queueing for presents from 1 heap

Parallelism: 7 kids getting 7 labeled presents, no queue

I have learned concurrency mechanism in this session. It is one the core feature of GO, very useful and we can use the standard library packages rather than using any additional packages like a TPL and so on, with simple keyword we can execute multiple function concurrently using the keyword “go” calling a routine, so the go routines will be running independently and automatically, this will increase the performance as well, here the channel is playing the important roll to synchronize the go routines.

Go with Interfaces

Go provide nice way of interface and inheritance concepts like we achieve the same thing in other programming languages like C#, which means I would not say Go programming language is object oriented programming languages.

Comparing to other Object Oriented languages. Go is much simpler, for defining the interfaces and types and implementing those interfaces by concrete methods on respective struct, so the compiler will check who are all implementing those methods from the Interface, they can have an access to the interfaces. Its bit difficult to grasp or understand the things who came from traditional object oriented programming.

Go With Web

We can create web application using standard library package called “http” and some sub packages as well, here the http middle ware stuffs is interesting to manage the shared functionalities, i.e. log the application trigger or any other authentication mechanism in the real world application and so on. JWT is also important concept to understand the token based message exchanging mechanism of JSON data across the client and server. There are lot of built in packages available to extend the application and customize more.

Unit testing

This section was really interesting for writing unit test for our application, here we are simply using the “testing” shared library packages and start writing the unit testing, for the assertion we can use Omega packages, but we don’t want to mentioned the packages as well, there are lot of interesting points in the unit testing, because it provides some interesting things like “Benchmark” test, it is really useful to benchmark the flow and produce the average result across different scenario, another interesting things is we can run the unit tests in parallel and can skip some long running unit tests using a flag by command line.

Conclusion

Overall the session was really useful to understand the concept of new programming language and to understand why its differ from other languages and so on, for me the interesting thing was concurrency, channels, interfaces, inheritance, unit testing, packages, so overall this is good for fresher’s compare to the other experienced developers, because they can quickly understand, for a C# developer it will take some time to understand the changes.

Thanks to ***Mr. Siju Varghese, Solution Architect who specializes in Go cloud and Docker.***