IPT – Intellectual Products & Technologies Ltd.
Bl. 44- office 2, Slatinska Str., Sofia 1574, Bulgaria
☏ (+359) 889 895196 ✉ office@iproduct.org http://iproduct.org/

Golang Programming

Course domain:	**Software Engineering**
Number of modules:	**1**
Duration of the course:	**36 hours**

Sofia, 2019

IPT – Intellectual Products & Technologies Ltd.

Bl. 44- office 2, Slatinska Str., Sofia 1574, Bulgaria
☏ (+359) 889 895196 ✉ office@iproduct.org http://iproduct.org/

Golang Programming

STUDY PLAN

Module name	1. Golang Programming
Lectures, astr. hours	18
Laboratory exercises, astr. hours	18
Total, astr. hours	36

Lecturer: ***Trayan Iliev***

IPT – Intellectual Products & Technologies Ltd.
E-mail: tiliev@iproduct.org

Target audience: Junior level developers with good algorithmic thinking. Practical experience with some object-oriented programming language will be an advantage.

Course duration: The total duration of the course is 36 hours + preselection test (2 hours).

Key takeaways:

- you will learn all major Go language constructs and recommended usage patterns;
- acquire practical experience in development of solutions to algorithmic problems, web clients and servers, web crawlers, REST APIs with Go;
- develop your own project – business application with web interface and database, which you can demonstrate as part of your project portfolio;
- develop *reliable and robust* Go applications by employing appropriate *error handling* techniques;
- manipulate files and resources – listing, walking, searching, reading, writing, appending, etc.;
- implement data persistence uisng SQL (MySQL) and NoSQL (MongoDB) databases;
- develop high-performance, concurrent applications in with *goroutines* and *channels*.
- correctly implement concurrent access to shared data using appropriate patterns and primitives – *sync.Mutex, sync.RWMutex, sync.Once*, etc.
- manage project dependencies using Go modules;
- implement generic containers using code generattion.

Course Program:

1. **[04.01] Introduction to Go –** history, why Go?, main features, installing Go, environment setup, hello-go program, Go programming and debugging in VSCode, git workflow setup, Go basic syntax. (3 h)
2. **[05.01] Program structure, data types, operators, control-flow statements**, **functions** – names, declarations, variables, constants, assignments, pointers, type declarations, packages, scopes, arithmetic, relational, logical, bitwise, assignment, type conversion, and other operators, decision making, loops, functions, value and reference parameters. ***Homework 1 (algorithmic problem)***. (3 h)
3. **[11.01] Composite types, functions, error handling** – arrays, slices, maps, ranges, structs, JSON, Text and HTML Templates, function declarations, recursion, multiple return values, errors, error handling strategies, function values, anonymous functions, closures, variadic functions, deferred function calls, panic, recover. (3 h)
4. **[12.01] Methods** – methods, method declarations, methods with pointer receiver, composing types by struct embedding, method values and expressions, encapsulation, examples. ***Homework 2 (Github JSON client)*** (3 h)
5. **[19.01] Interfaces** – interfaces as contracts, interface types, interface conformance and duck typing, interface values, sorting with sort.Interface, http. Handler interface and routing with ServeMux, error Interface, type assertions, discriminating errors using type assertions, querying behaviors using interface type assertions, type switches, examples, best practices using interfaces in Go. ***Homework 3 (Building a simple web application)*** (3 h)
6. **[25.01] Goroutines and Channels** – goroutines, concurrent service implementation examples, channels, parallel loops, concurrent web crawling example, multiplexing with select, goroutines cancelation, concurrent directory processing example. (3 h)
7. **[26.01] Concurrency with Shared Variables** – goroutines and threads, data races, mutual exclusion - sync.Mutex, read/write mutexes - sync.RWMutex, memory synchronization, lazy initialization - sync.Once, Go race detector, examples. Chat server example. ***Homework 4 (Concurrent Web Scrapping)*** (3 h)
8. **[01.02] Working with SQL and NoSQL databases** – using database/sql (sql.DB), importing MySQL DB driver, accessing database, retrieving result sets, modifying data and using transactions, using prepared statements, handling errors, connection pooling, best practices. Using MongoDB and MongoDB Go Driver, connecting, using BSON objects in Go, CRUD operations. Building a sample web application with MongoDB – *GoBlogging*. (3 h)

9. **[02.02] Building RESTful APIs and web clients with Go** – practical lab. ***Choosing course projects.*** (3 h)
10. **[08.02] Modules and dependency management using go mod –** modules, go.mod file, version selection, semantic import versioning, installing and activating module support, defining a module, upgrading and downgrading dependencies, daily workflow. ***Projects mentoring***. (3 h)
11. **[09.02] Code generation. Summary** – generics and their purpose, type-specific wrappers with type-agnostic implementations, creating a generator, creating a template, running gofmt and goimports, alternatives – using interfaces instead of types. Q&A. (3 h)
12. **[15.02] Projects demonstration.** (3 h)

Presentations, demonstrations, and hands-on exercises during the training are conducted in parallel in order to achieve immediate reinforcement of conceptual knowledge in practice. The training groups are small allowing for individual mentoring of each participant by the instructor. There will be multiple homeworks, as well as final project demonstration. During the training there will be opportunity for discussion of additional questions the participants are interested in.

Resources

1. The Go Programming Language, Donovan, A., Kernighan, B., Addison-Wesley, 2016
2. Get Programming with Go, Youngman, N., Peppé, R., Manning, 2018
3. Official Go programming language documentation website - https://golang.org/doc/