Introduction to GO

CS240

20/8/2017

Course overview

- Course website: http://web.kaust.edu.sa/Faculty/MarcoCanini/classes/CS240/F17/
- Piazza: https://piazza.com/kaust.edu.sa/fall2017/cs240/home
- Grades are based on lab assignments and exams
- We will use Go as the programming language for the assignments
- Prof. Canini is away this week; he will expand more on the class overview next week
- Today we will re-introduce version control and go over the basics of Go (golang)

Version control

It is a way to manage different versions or revisions of the files.

Examples:

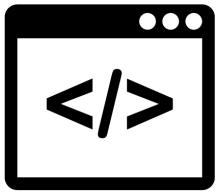
- Undo/redo buffers
- Google docs
- Overleaf (online Latex)
- Multiple versions

```
sibyani@kaust:~$ ls
assignment1-1.go
assignment1-2.go
assignment1-3_work_in_progress.go
```

Example:

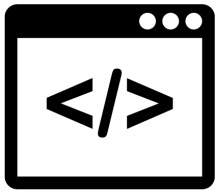
sibyani

finished 1-1



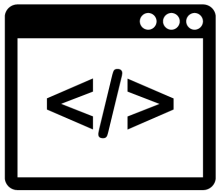
assignment1.go

finished 1-2



assignment1.go

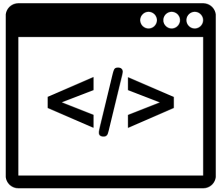
working 1-3



assignment1.go

finished 1-1

sibyani

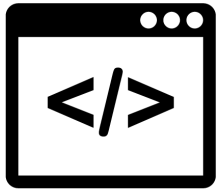


assignment1.go

github/sibyani

sibyani

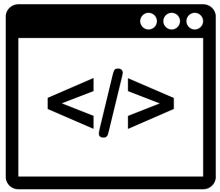
finished 1-1



assignment1.go

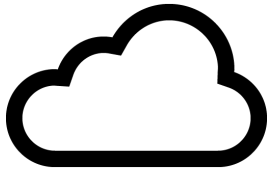
github/hsibyani

finished 1-1



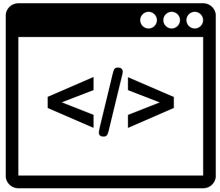
assignment1.go

sibyani

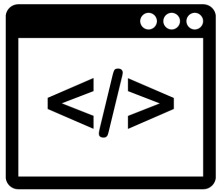


github/hsibyani

finished 1-1

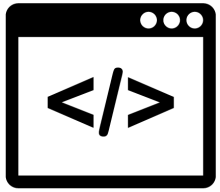


assignment1.go



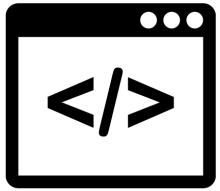
assignment1.go

finished 1-2



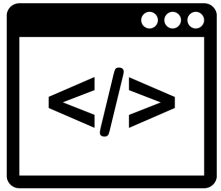
assignment1.go

finished 1-2



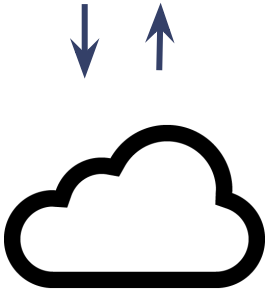
assignment1.go

working 1-3



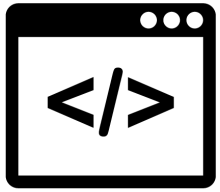
assignment1.go

sibyani



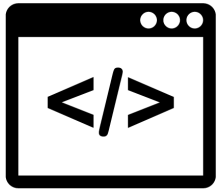
github/hsibyani

finished 1-1



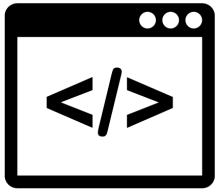
assignment1.go

finished 1-2



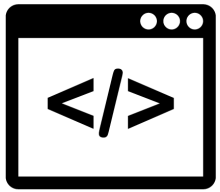
assignment1.go

working 1-3



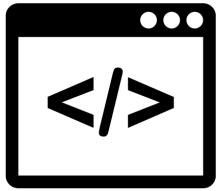
assignment1.go

finished 1-1



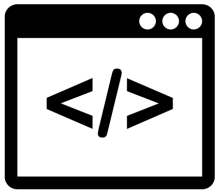
assignment1.go

finished 1-2



assignment1.go

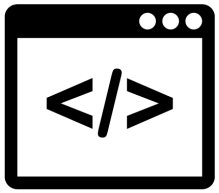
working 1-3



assignment1.go

finished 1-1

sibyani

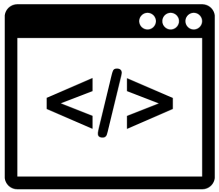


assignment1.go

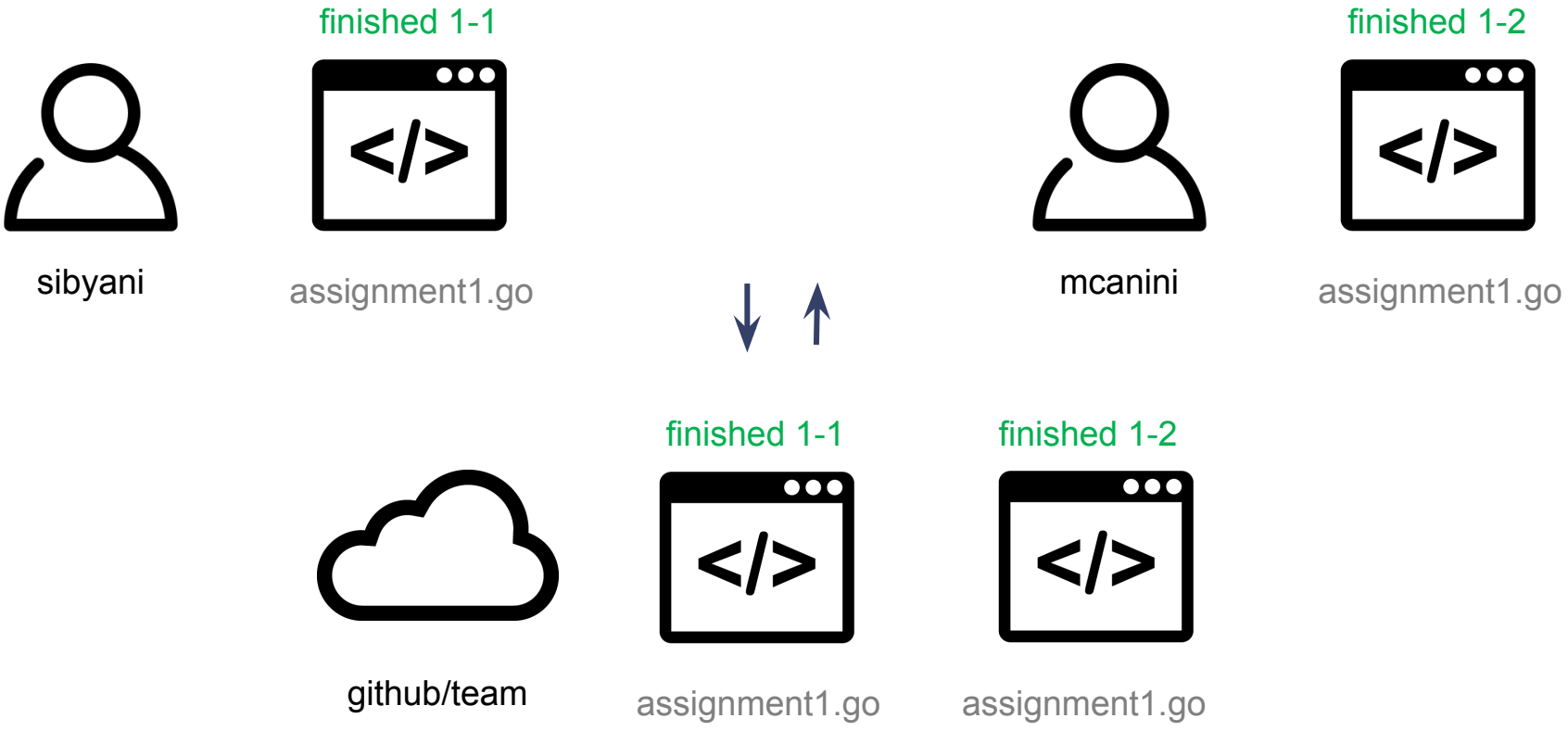
mcanini

finished 1-1

github/team



assignment1.go


finished 1-1
sibyani
assignment1.go
finished 1-2
mcanini
assignment1.go
finished 1-1
finished 1-2
github/team
assignment1.go
assignment1.go

Git

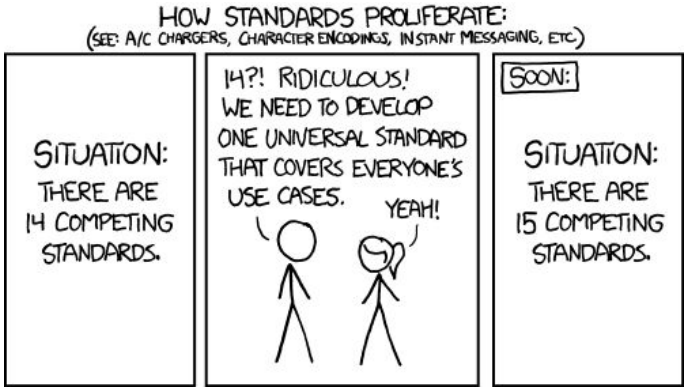
- Git is a version control system (not the only one)
- Most common commands
  - `git init`: create git repository
  - `git add`: add a file to be tracked in the repository
  - `git commit`: commit the current tracked files
  - `git push`: push the local files to the cloud
  - `git pull`: pull the cloud files locally
- Start with git at: https://try.github.io

GitHub
Student Developer Pack

https://education.github.com/pack

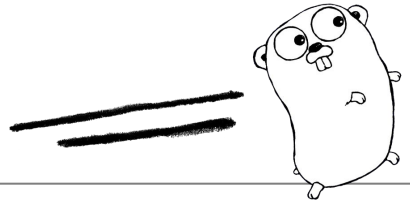
Why Go?

- Simpler, cleaner, and faster programming
- Fast compilation
- Easier interface abstractions
- Garbage collection
- Concurrency as a goal
- Read more in: https://golang.org/doc/faq#Origins



source: xkcd

Who uses Go?



CLOUDFLARE

UBER

Dropbox

Google

and many more ...

Installation

- Usually straightforward (package manager or source)
  - `apt` (Ubuntu)
  - `yum/dnf` (RedHat)
  - `brew` (macOS)
- Pay special attention to environment variables
- You may need to make sure you configure some of them (GOPATH and GOROOT)

Demo: Basics and declarations

Demo: Packages

Demo: Control flow

Demo: Extras

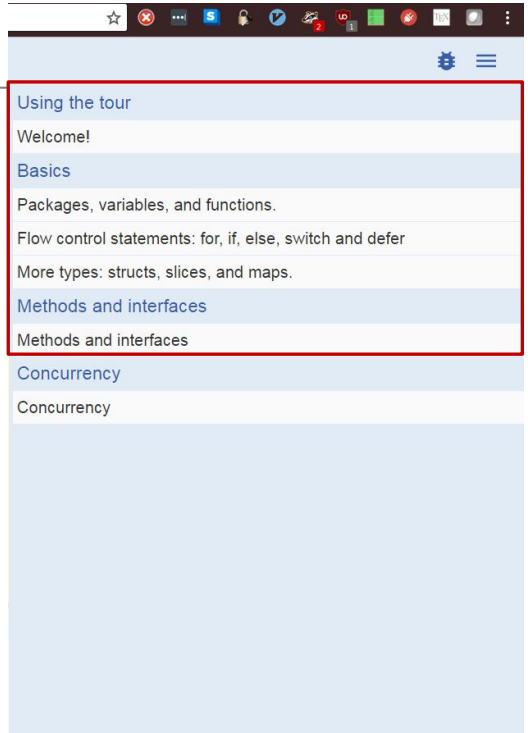
To do before next class

Join Piazza

Install Go

Go tour – up to the concurrency: https://tour.golang.org

Do some exercises

And if you have time:

- How to write Go code: https://golang.org/doc/code.html
- Effective Go: https://golang.org/doc/effective_go.html