S01 Software Development Tools

IDEs
Revision Control
Gitlab and Git

ANU – School of Computing – Structured Programming 1110 / 1140 / 6710

Integrated Development Environments

- An editor to do more than just *write* code.
  - Syntax highlighting, completion, continuous compilation, testing, debugging, packaging
  - Code analysis and refactoring capabilities
- Examples: Eclipse, IntelliJ, VisualStudio, XCode

Version Control (VCS, RCS, SCM)

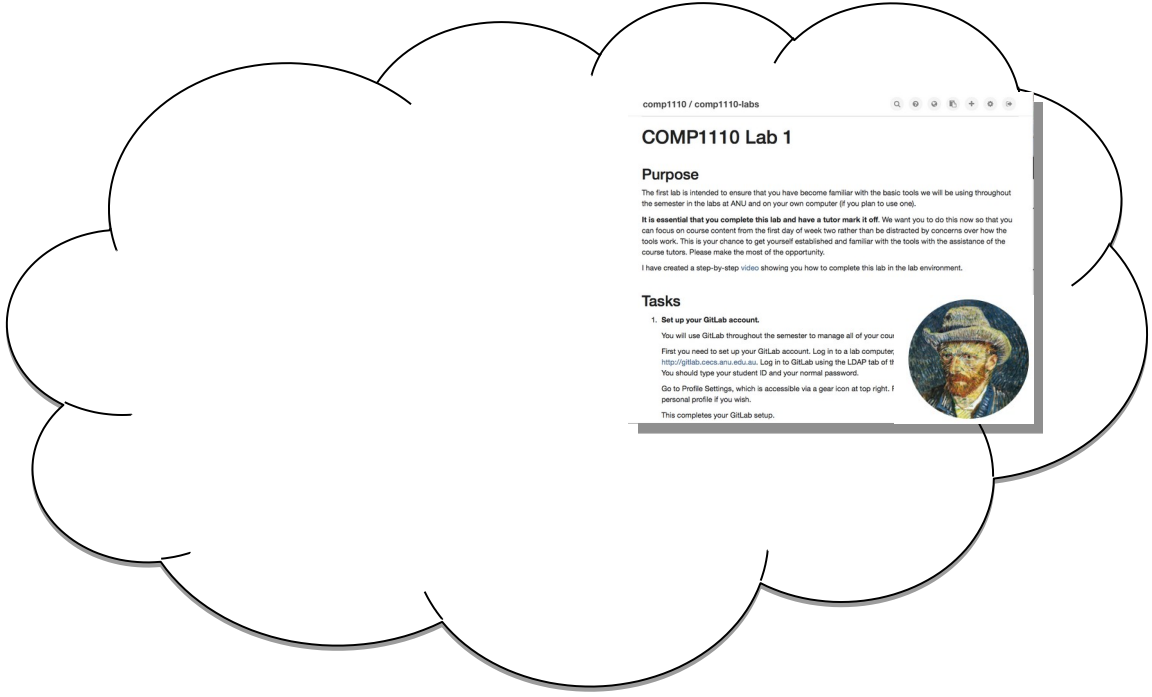
- Indispensable software engineering tool
- Solitary work
  - Personal audit trail and time machine
  - Establish when bug was introduced
  - Fearlessly explore new ideas (roll back if no good)
- Teamwork
  - Concurrently develop
  - Share work coherently

Git & Gitlab

- Distributed version control system
  - hg, git, and others
- Contrast with centralised version control
  - cvs, svn, others

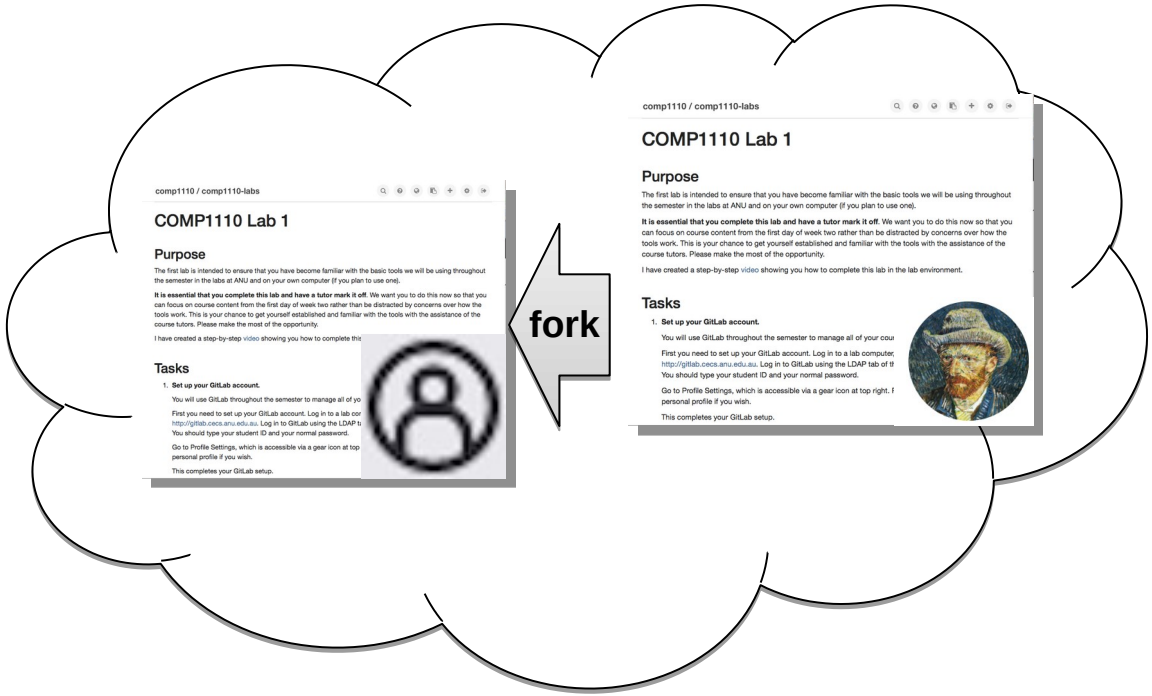
We will use a distributed version control system – git – and a server – the ANU teaching gitlab – for sharing and submitting course work.

Git & GitLab



(master) labs repo
(owned by comp1110)

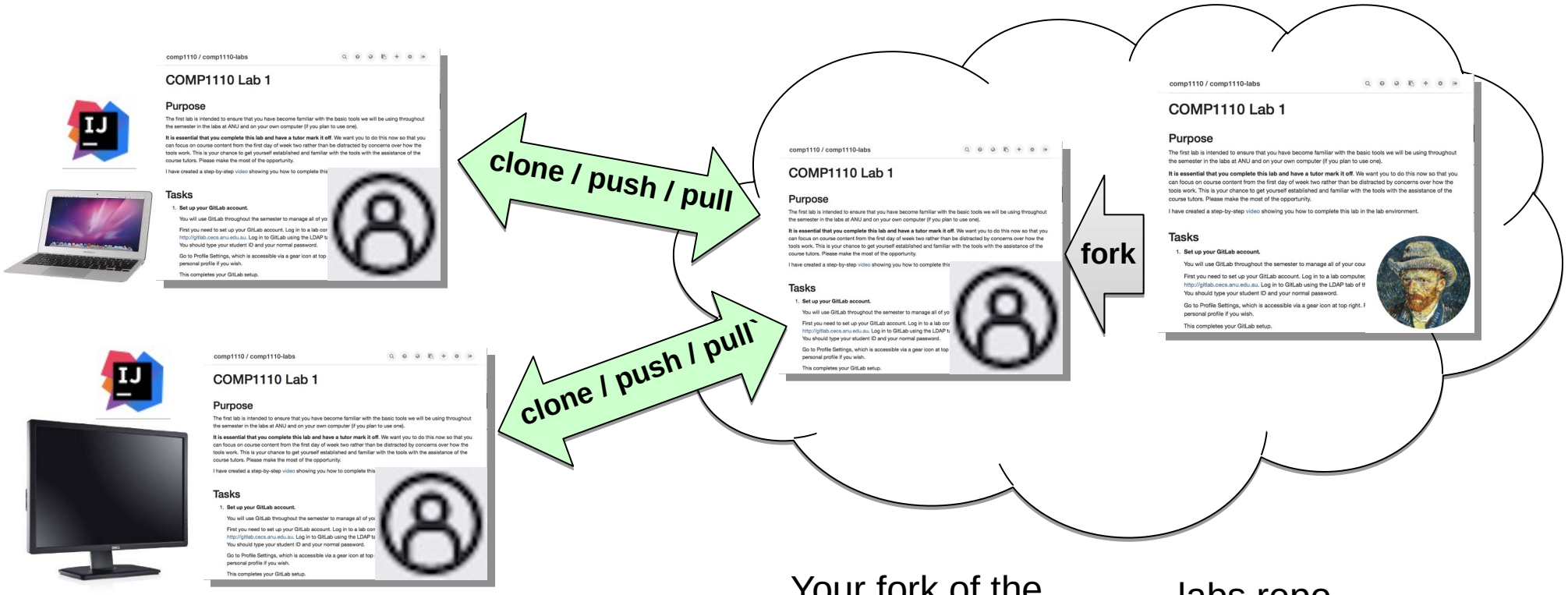
Git & GitLab



Your fork of the labs repo (owned by you)

labs repo (owned by comp1110)

Git & GitLab



Clone(s) of your fork of the labs repo
(owned by local user)

Your fork of the
labs repo
(owned by you)

labs repo
(owned by comp1110)

Recap

- **Repository** (“repo”): A copy of a project and its history.
- **Gitlab**: A server (remote) that stores repos
  - ANU teaching gitlab: https://gitlab.cecs.anu.edu.au
- **Clone**: A working (local) copy of a repo.
- **Pull**: Fetch updates from a remote to a working copy.
- **Push**: Send updates from a working copy to a remote.
- **Commit**: An update to a repo.

IntelliJ Git Integration

- Clone an existing repository:
  - “Get from VCS” on splash screen
- Other operations:
  - Git menu
  - right mouse click > Git