

Week	Topics
1	CLI navigation. Common commands (ls, grep, top, sed, find). Vim. Build tools Practical: Create a folder on your Desktop, create a text file in it, open any website and copy its text into your file, replace occurrences of some word into another, save - all via CLI.
2	IDE overview. IDE vs CLI. Version control and Git. Compare to GitHub. Basic Git commands demo via IDE or terminal Practical: Set up your GitHub and create repositories for this and other courses. Submit first files (test or anything related to the course).
3	Scripting (Bash or Python) for automating tasks. Practical: Write a script to automatically push to your GitHub repo every new file appearing in your local directory (related to this course)
4	Debugging via IDE. Breakpoints, stepping, logpoints, data inspection Practical: Add breakpoints into your OOP code and step in/out of your functions and inspect changing variable values.
5	Types of tests (unit testing, integration testing, end-to-end testing). Testing Frameworks. Writing tests. Common test cases to consider. Practical: Cover your one of your OOP class codes with passing tests AND submit to your GitHub
6	How Internet works. Practical: Inspect one of your favorite websites internal structure. Write a simple web app to practice your skills.
7	HTML, CSS and JS. Practical: Improve your simple web app to practice your skills.
8	1. Networking basics (Python): client/server, TCP/IP, HTTP. Status codes. Practical: Build your own client/server. Extra: do it in Java
9	2. Networking basics (Python): Writing client-server. API development: Protocols (HTTP). Data formats (JSON, XML). API styles (REST, RPC). HTTP methods. Practical: Call one of popular APIs, like Google Maps, and go over it.
10	3. Networking basics (Python): API design and development. API endpoints and resources. Best practices: Auth. Versioning. Documentation. Errors and exceptions. ?Frameworks Practical: Reusing your own client/server add REST APIs on top of it. Extra: Add documentation.
11	4. Networking basics (JS): Let's write REST APIs using ExpressJS framework Practical: Implement other methods of the app.
12	1 Databases. Using SQL to read/update? Connecting APIs to databases. Deploying and hosting APIs. Practical: Given a test db, query specific things ? Add a db into your client+server+API application, update your APIs and tests. Extra: now host it all
13	Freestyle