- Classes / Cpp / header files / Forward declaration
- Makefiles
- Version/source control
- Testing
- Testing principles (systematic – early and often – automatic)
- Testing your program (dealing with lists, files, strings, …) / how to divide your space into representatives.
- Testing your program with i/o streams
- How to design your program - design recipe
    - Determine the **representation**
    - Write a **purpose** statement
    - Write **examples**
    - Create a **template** for the implementation
    - Finish **body** implementation case-by-case
    - Run **tests** (Fix if tests fail)
- Debugging (difficulties in debugging – debugging strategies – debugging methods - how to use lldb tool to add/remove breakpoints, check frames,..)
- Defensive programming (where to use assertions / where to use error handling – error types, error handling techniques)
- Test/code coverage (line vs branch vs path vs function coverage)
- Documentation (coding style – keys to effective comments)
- Power of variables
- Libraries
- how to use -I flag
- test automation (what do we need, when to (not) use it, benefits)
- design patterns