Applications

- Apps should have **persisent** data, meaning that it will still exist even if the power is shut off.
  - You tend to see persistent data in secondary storage - on disk or stored on servers (cloud)
- Good applications should be efficient (obviously) but also understandable to both the user and developer alike
  - On the note of being understandable to developers, the "holy grail" of applications is one that is **extensible**, meaning that a developer can easily view the source code and extend additional, customizable features for the applications (think Emacs via ELisp)
- Different applications should not be able to directly communicate with each other; the operating system should act as a middleman (i.e. one application uses the OS to create/edit a file, which is then viewed by another application)
- Applications can benefit from **introspection**, in which a program can look at itself (i.e. its own state) - a concept of which is often vital when debugging applications.

Software Tools

- A complex problem should not be solved by a single, large program tool. Rather, it is optimal to create multiple, smaller "tools" that can solve simpler problems and combine towards tackling the larger problem at hand.
  - A web application, for example, is not written in an entire Javascript file - it is split into multiple "modules" that may handle different tasks (Database model handler, Router, React for frontend view, etc.)

Little Languages

- Software tools (as aforementioned) often require configuration or extensibility. This should be handled with a "smaller" language tailored towards this specific tool rather than a larger language.
  - Consider Lisp for Emacs or sh/bash for Linux.

Layering

- An often useful pattern in software construction is layering - meaningful functionality is often "layered" on top of each other.
  - Scripting languages, for example, often have an interpreter written in a low level language (C/C++), which is then layered with an interface for the user.
  - Layering is even encountered with the Internet Protocol Suite