**Design ideas**

- encapsulation

- prime directives: no surprises, maximize cohesion, minimize coupling, keep it small and simple

- distinguish between application-side and solution-side

- use namespaces to define a project taxonomy

- Project name or Source, Task names, Data type names, e.g.,

CSE687.Parser, CSE687.Tokenizer, CSE687.BlockingQueue

- top down (emphasis on partitioning)

- bottom up (emphasis on implementation) design

- partitioning: classes, packages, processes, and machines

- distinguish between interface, abstract class, and concrete class

- component: interface, object factory, concrete implementation, dll package

- delegation and events

- subscription: the observer pattern

- asynchronous messaging

- declarations: types, values, and processing ??

**Design notes**

- software size matters

- package structure is important

- Favor local consequences

- Extract function instead of cut and paste

- display information, not data

- distinguish between test types: construction, unit, and regression

- Power and perils of sharing

- try to avoid distributing the definition of control across packages ??

**Language notes**

- distinguish between value and reference types

- RAII

- initialization

- Vfptbl