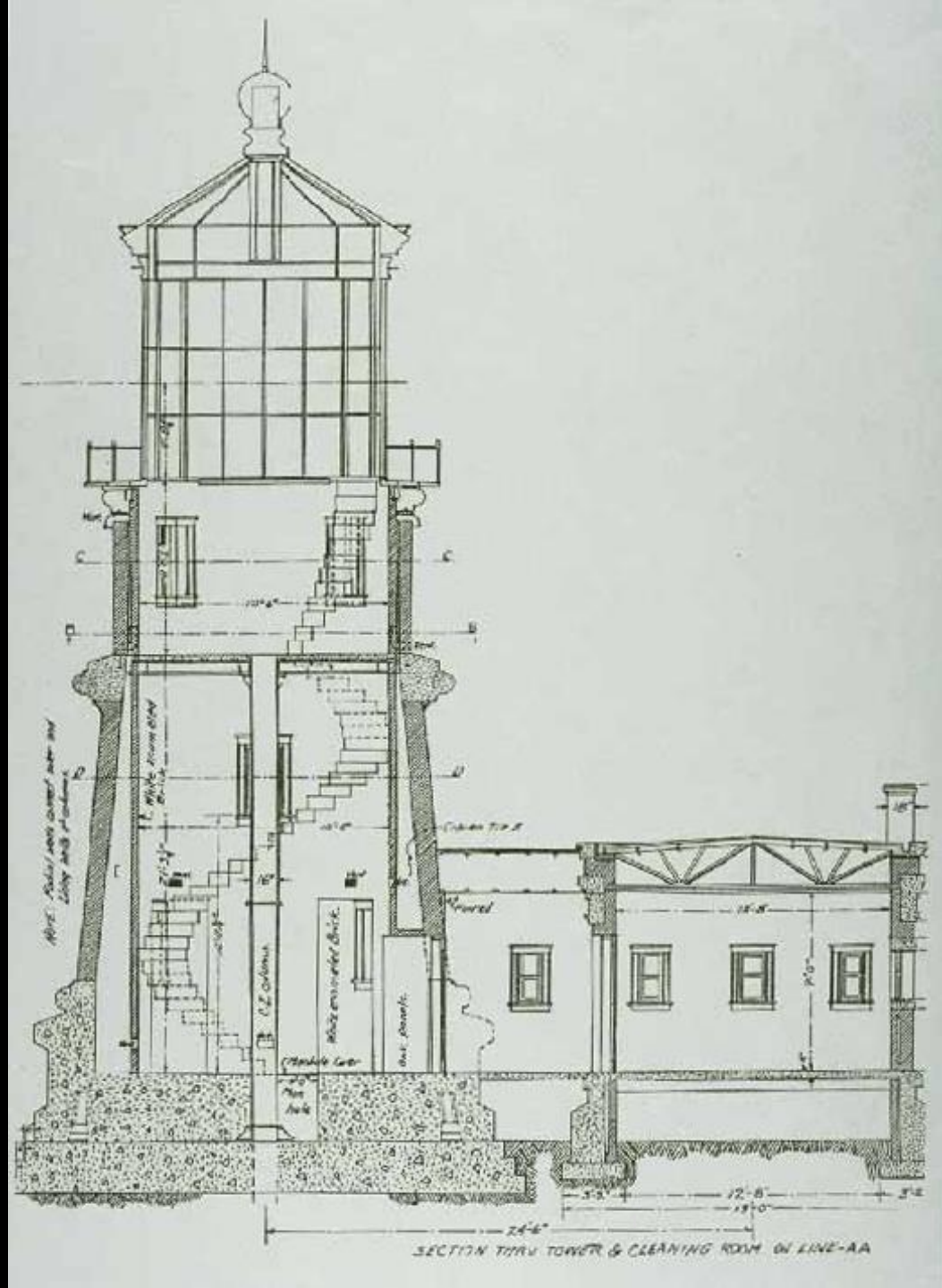
Software Architecture & Design



Object-Oriented Design

Revising the basics

Object Oriented Primer

Revising The Basics

OO Modelling Concepts

- Class
  - i.e. Concrete Class
  - Typically a Noun
  - Represents an entity
- Object
  - An instance of a class
- Method
  - Function scoped within a class
- Member variables
  - Encapsulated data (static/variable)
- Interface
  - Declaration of behavior
  - Separate from implementation
- Abstract Class
  - Default behavior
  - Default implementation of interfaces
  - Common/Shared implementation
- Template
  - Blueprint for a class
  - Typically vary by types, not behavior

OO Design Principals

- Inheritance
- Encapsulation
- Polymorphism
- Message Passing

OO Design Principals

- Inheritance
- Encapsulation
- Polymorphism
- Message Passing

- Things' relate to other 'things'
- Relationship
- Two types of inheritance
  1. 'is a' – e.g. Triangle 'is a' shape
  2. 'is implemented in terms of'
- Allows you to extend behavior or override behavior
- Treat similar things in a generic way

OO Design Principals

- Inheritance
- Encapsulation
- Polymorphism
- Message Passing

- Represents ‘has a’ relationship
- Enables you to hide details, data/properties or complexity
- Enables homogeneity
  - Allows you to treat different things in the same way

OO Design Principals

- Inheritance
- Encapsulation
- Polymorphism
- Message Passing

- 'Fancy word' to mean treat different things generically (i.e. in the same way).
- Classes sharing the same interface can be interacted with generically.
- Treat a collection of 'shapes' uniformly/generically.
  - e.g. collection containing squares, circles, triangles, parallelograms, etc. and request to 'print'
- Achieved by separating interfaces from implementations

OO Design Principals

- Inheritance
- Encapsulation
- Polymorphism
- Message Passing

- Somewhat abstract concept in OO design
- Practical implementation = method invocation
- More of a convention?
- Relates to delegation & notification
  - Call another object to do work
  - Notify another object/bus
  - Will be called when an event occurs

Inheritance versus Encapsulation

- Modelling
- Relationship = ‘is a’
- Relationship = ‘has a’
- e.g.
  - A motor car ‘is a’ vehicle
  - A motor car ‘has a’ engine

Q&A

Discussion Time

Thank you

Recommended Reading