CS Concepts

Paradigms

Procedural
- Procedure
- Record/Struct
- Control Flow
- Memory

Object-Oriented
- Method
- Object/Class
- Inheritance
  - Composition
  - Delegation

Functional
- Function
- Interface/Type

General Concepts

Lower Level
- Primitives
- Scope
- Value vs. Reference (pointer), Mutability
- Call-by-value, call-by-name, lazy
- Modularity, separation of concerns
- Recursion
- Serialization/Encoding/Representation
- Syntactic Sugar
- Testing

Higher Level
- Complexity Theory
- Data Structures
- Design Patterns
- Reflection
- Parallelism
- Event-Based, IO

Engineering

Lower Level
- Debugging
- Readability
- Iterative Approach
- Documentation
- Version Control
- Frameworks
- Code Review

Higher Level
- Test-Driven
- Design (Interfaces, Architectural)
- Robustness
- Collaboration

Basic Coding

Languages

Scala 2 — all must learn
C#
Python
Typescript
Go
Rust (Hard)
C (Hard)
Haskell

Exercises

Modular Arithmetic
Fibonacci: LW/ timing comparison)
Set Intersection (Join)
π computation
Mandelbrot, Fractal fun

Search
Sort
Change-of-base
ModExp (square + multiply)
Euclid's Algorithm (Bézout), Modular Inversion
Big Int
Conway's Game of Life
Polynomial Interpolation
Elliptic Curve Addition, Multiplication
Rogue Key Attack Impl
Reused Nonce Key Derivation Impl
Counting Paths on Graphs
Normal form of group element given presentation

Voting simulation
Merkle Tree Chunks

Wagner Collision-Finding Impl
Convex Hull for finite set
Primality Testing
Diffy-Q simulation

Graph Algos
Graph Coloring (Backtracking search)
Schnorr Sigs, Adaptor Sigs, Batch Verification
Anything From projecteuler.net/archives

Git / GitHub

Terminal

Tutorial

Shared Cheat Sheet / Knowledge Blurbs organized by domain on GitHub

- Primitives
- Scope
- Syntax
- Syntactic sugar
- Mutability
- Types
- Organizational / modularity tools
- Memory interface
- Installation
- Testing
- Style Guides