

Computer Science Introductory Course Topics

CS 111

- Basics of programming
- Types and type signatures
- Unit testing, test-driven development
- Recursion and special cases of recursion (iterative, generative, tree recursion)
- Debugging and reasoning about program failure
- Basic compound data (lists and record types)
- Procedures as data
- Functional programming (lambda expressions, etc.)
- Functional list and stream processing (mapping, filtering, folding)
- Imperative programming (sequencing, assignment, object mutation)
- Packaging code with data (e.g. methods in classes)
- Run-time type dispatch

CS 211

- Semantics of C and C++
- Basics of OOP - classes, inheritance, visibility, etc.
- von Neumann machine model (memory, addresses, etc.)
- Pointer semantics, pointer arithmetic, etc.
- Basics of the Unix tool chain
- Reasoning about and debugging in languages that aren't type- or memory-safe, i.e. memory leaks, uninitialized pointers, buffer overflow, etc.

CS 214

- Asymptotic performance analysis (best, worst, and average case)
- Amortized analysis
- Basic sequence structures (arrays, dynamic arrays, linked lists)
- Tree representation
- Tree walks
- Search trees, including self-balancing trees (we don't expect you to memorize the self-balancing algorithms, though!)
- Hash tables and hash sets
- Hash function design
- Sort algorithms
- Graph representations
- Graph walks
- Shortest path algorithms for non-negative edge costs
- Dynamic programming (mostly just for Floyd-Warshall)
- Representing disjoint set partitions, particularly the performance of the union-find algorithm with path compression and union-by-rank.