Final Review (1)

- Introduction to testing
  - Basic concepts: fault, failure, error, test case, testing, debugging, verification & validation
  - The testing process: test generation, test execution, and test evaluation
- Input space partitioning
  - Equivalence partitioning (interface-/functionality-based approach), boundary-value analysis
- Combinatorial testing
  - Combinatorial explosion, t-way testing, pairwise testing, the IPO algorithm

Final Review (2)

- ❑ Graph-based testing
  - Basic concepts: path, simple path, prime path, test path, tour, sidetrip, detour
  - CFG: basic block, node coverage, edge coverage, prime path coverage
  - DFG: definition/use, du-pair, du-path, all-defs/all-uses/all-du-paths coverage
- ❑ JUnit
  - Assertions, test fixtures, test runners

Final Review (3)

- ❑ Test Data Generation
  - Symbolic execution, constraint solving, search-based strategies
- ❑ Predicate Testing
  - Basic concepts: predicate, clause, active clause
  - Coverage criteria: predicate coverage, clause coverage, GACC/CACC/RACC
- ❑ Mutation Testing
  - Program-based mutation testing, mutant, reachability/ infection/propagation, mutation operators

Final Review (4)

- ❑ Regression Testing
  - The RTS problem, test revalidation, test selection, test minimization, test prioritization
- ❑ Security testing
  - Security vs functional testing, input validation, buffer overflow/command injection/XSS, fuzzing
- ❑ Software maintenance
  - Maintenance vs development, software change, process models, program understanding, reverse engineering, configuration management, management issues

Final Review (5)

- ❑ Code Review
  - What, why, when, and who
  - Lightweight vs formal review
  - Practical tips, tool support
- ❑ Version Control
  - Product space, version space, version delta (embedded vs directed), externsional vs intensional versioning
- ❑ Software Refactoring
  - What and why, presentation vs computation, code smells