SE 433 Final Exam Review Guide:

All materials covered in the lectures will be included as potential exam questions. The questions will **definitely** include topics that we have covered since the midterm, but may also include topics from the first half of the course. (I.e. there will be bias toward untested material from the last few lectures.)

#### Exam format:

Expected duration: 1½ hours.

Maximum time allowed 3 hours.

**Reading:** Text Chapters 1-5, 9-12, 17-18, 20-24

**See also**: ISTQB Sample Exam Questions: [http://www.testingexcellence.com/istqb-quiz/istqb-sample-exam-questions-testing-fundamentals/](file://localhost/Software%20Testing%20Fundamentals%20Online%20Exam/%20http/::www.testingexcellence.com:istqb-quiz:istqb-sample-exam-questions-testing-fundamentals:)

###### **Topics covered**

**Kinds of testing**

* Black box
* White box
* Functional
* Structural
* Statistical

**Static analysis**

* Difference among and definition of Coverage and Adequacy criteria
* Statement
* Branch
* Condition,
* Branch-condition,
* Basic condition coverage
* Branch and basic condition coverage
* Compound condition
* MC/DC adequacy criterion
* Paths
* Boundary/interior
* Loop boundary
* LCSAJ
* Cyclomatic, basis path
* Test suite for each above
* Control Flow Graphs
* Test paths
* Test cases
* Test coverage

**Functional Testing**

* Development of test cases
* Equivalence partitioning test technique
* Equivalence partitions are needed
* Types of equivalence partitions
* Weak normal
* Strong normal
* Weak robustness
* Strong robustness
* Boundary tests

**Types of testing**: definition, use/purpose, verification and/or validation

* Unit Testing
* Regression testing
* Acceptance testing
* System testing
* Integration testing

**Integration test strategies**

* *Top-down* strategy
* *Bottom-up* strategy
* *Big Bang*
* *Sandwich*
* Given a system, describe sequence of integration and test for a set of modules

**Concurrency**

* What is a *race condition*?
* Why concurrency related defects, such as race condition, are difficult to discover in testing.

**Pair-programming (Ch. 18.5)**

* What is pair-programming and why pair-programming can produce better quality code?

**Tools**

* JUnit, ANT, Eclipse, Cobertura

**Measuring software progress**

* Milestones and status reporting

**QA, Testing, Measurements**

* Why we take metrics
* Cyclomatic complexity
* % Coverage
* Availability
* MTBF
* MTTR
* Defect Metrics
* Quality tools

**Misc.**

* Benefit of independent testing
* Relevant cost of defect detection in various phases
* Purpose of or valid goal for a test team