# SE 433 Midterm Review Guide

All material covered in lectures 1 to 5 may be included in the Midterm exam

### Exam format:

Expected duration: < 1.5 hour.  
Maximum time allowed 3 hours.

### Review

You did your readings, yes?

* Text & class slides
* Assignments

### Lecture topics:

Introduction to Software Testing

Software Quality

Eclipse

Unit Testing

JUnit

Black Box Testing

Ant

### Topics covered:

1. Difference between Validation and Verification
2. **In testing,** (JUnit) difference between: An error, A failure, and A defect
3. The purpose of software testing.
4. Types of testing:
   1. Unit testing
   2. Integration Testing
   3. System Testing
   4. *Alpha* testing
   5. *Beta* testing
   6. *Regression* testing
   7. Acceptance (Release testing)
5. Testing methodology
   1. Functional testing, i.e., black box testing,
      1. Equivalence partitions
         1. Weak normal test
         2. Strong normal test:
         3. Weak robustness test
         4. Strong robustness test:
      2. Boundary values
   2. Structural testing, i.e., white box testing,
   3. *Random* testing
6. How Exhaustive can/should testing be.
7. Measures of system quality
8. System availability, MTTR, MTBF/MTTF
9. Software quality factors
   1. Efficiency
   2. Performance
   3. Reliability
   4. Usability
   5. Maintainability
   6. Portability
   7. Interoperability
   8. Safety
   9. Security
   10. Scalability
10. In scenarios concerning unit test cases using **JUnit**. Determine the verdict of each test case:
    1. Pass
    2. Fail
    3. Error
11. Functional testing: Given a set of inputs
    1. Identify the valid and invalid equivalence classes for each input parameter.
    2. Identify test suites with the *minimum* number of test cases that would satisfy each of the following coverage of equivalence classes.
       1. Weak normal test suite
       2. Strong normal test suite
       3. Weak robustness test suite
       4. Strong robustness test suite
    3. Add test cases using normal boundary values to the weak normal test suite
    4. Add test cases using robustness boundary values to the strong robustness test suite