Australian National University

S4

Test Driven Development

Test-Driven Development (TDD)
JUnit

Test Driven Development (TDD)

TDD “red, green, refactor

1. Create test that defines new requirements
2. Ensure test fails
3. Write code to support new requirement
4. Run tests to ensure code is correct
5. Then refactor and improve
6. Repeat

Key element of agile programming

Unit Testing & JUnit

Unit Testing – test small parts of your program individually

JUnit provies a framework to do this in Java

- Developed by Kent Beck ("extreme programming" movement)
- Integrated into IntelliJ
- Useful for
  - TDD (Test Driven Development)
  - Bug isolation and regression testing
    - » Precisely identify the bug with a unit test
    - » Use test to ensure the bug is not reintroduced

JUnit

Methods marked with @Test can be run as tests

When JUnit is called on a class, it runs all tests and generates a report (a failed test does not stop execution of subsequent tests)

JUnit has a rich set of annotations that can be used to configure the testing environment, including: `@Test, @Ignore, @BeforeEach, @BeforeClass, @AfterEach, @AfterClass, @Timeout`

Within tests, Assertions are used to actually check things. These are static methods like `assertTrue, assertFalse, assertEquals`