Unit Testing

CS 272 Software Development

Unit Testing
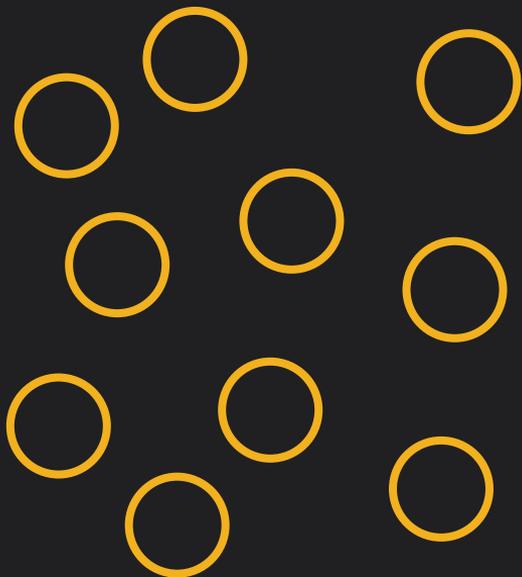
Integration Testing

System Testing

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO
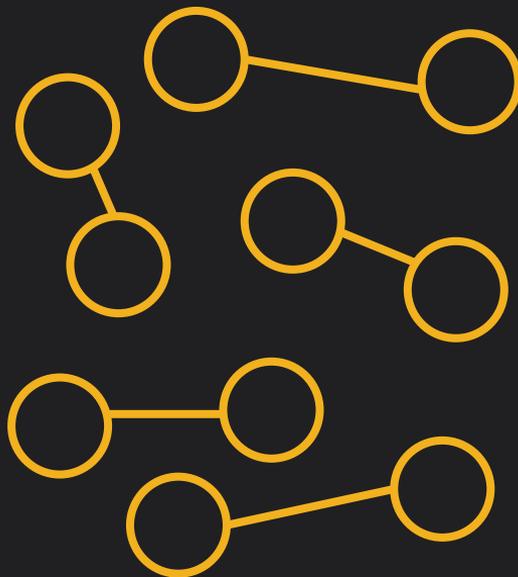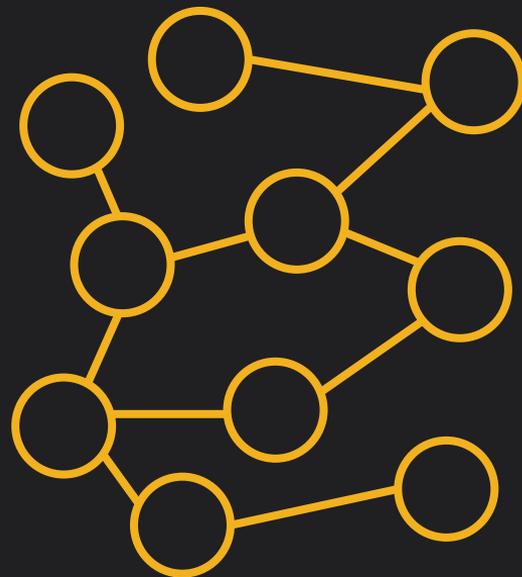
Unit Testing

- Tests **individual** methods

- Improves software **correctness**

- Testing is **automatable** and manageable

- Performed by **developers**

- Usually part of **continuous testing** and/or **test-driven development**

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

JUnit Framework

- Used for primarily for **unit testing**

- Built into Eclipse IDE (and most IDEs)

- Uses custom assertion methods (not **assert** statements)

- Latest versions (v4+) uses **annotations**
  - `@Test, @BeforeEach, @AfterEach`

https://junit.org/junit5/

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

JUnit Assertion Methods

- `assertFalse()`
- `assertTrue()`

- `assertNotNull()`
- `assertNull()`

- `assertNotSame()`
- `assertSame()`

- `assertArrayEquals()`
- `assertEquals()`

https://junit.org/junit5/docs/current/user-guide/#writing-tests-assertions
https://junit.org/junit5/docs/current/api/org/junit/jupiter/api/Assertions.html

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

JUnit Annotations

- Use **@BeforeEach** for code called before every test
  - Used to setup environment before test

- Use **@Test** to indicate a method is a unit test
  - Should contain one assertion method*

- Use **@AfterEach** for code called after every test
  - Used to clean up environment after test

https://junit.org/junit5/docs/current/user-guide/#writing-tests-annotations

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

JUnit @Test Annotations

- Many related test annotations
  - @Test, @RepeatedTest, @ParameterizedTest

- Double check importing right annotations!
  - Similar to annotations in JUnit 4
  - Imports should be from `org.junit.jupiter.api`

- Using `System.exit()` can break unit tests

https://junit.org/junit5/docs/current/user-guide/#writing-tests
https://junit.org/junit5/docs/current/api/org/junit/jupiter/api/Test.html

Advanced JUnit Tests

- Able to create **parameterized tests**
  - Create single test that expects parameters
  - Create specific list of parameters to test

- Able to **group tests** into nested classes
  - Create multiple nested classes of test cases
  - Can also inherit test cases from other classes

https://junit.org/junit5/docs/current/user-guide/#writing-tests-parameterized-tests
https://junit.org/junit5/docs/current/user-guide/#writing-tests-nested

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

Example Normal Test

```java
@Test
public void testNullFlag() {
    boolean actual = ArgumentMap.isFlag(null);
    Assertions.assertFalse(actual, "null");
}
```

https://github.com/usf-cs272-fall2021/homework-ArgumentMap-template/blob/9d6de268c89caf7a1b3bbd18c57f7304eaf e87e2/src/test/java/ArgumentMapTest.java#L89-L97

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

Example Parameterized Test

```java
@ParameterizedTest
@ValueSource(strings = { "-a", "-hello", … })
public void testValidFlags(String flag) {
    boolean actual = ArgumentMap.isFlag(flag);

    Assertions.assertTrue(actual, flag);

}
```

https://github.com/usf-cs272-fall2021/homework-ArgumentMap-template/blob/9d6de268c89caf7a1b3bbd18c57f7304eaf
e87e2/src/test/java/ArgumentMapTest.java#L57-L71

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO