CPSC-224 Software Development

Unit Testing

Yu Wang

wangy2@gonzaga.edu

March 19, 2025

Announcement

- Recruiting RA for summer break
- Team Form Result on Canvas
- Final Project Part1
- HW3 Uploaded on Canvas
- HW0 –
  - Career Dev 02 – Resume Revision
  - Career Dev 02a – I visited the ProRep Document

Daily Attendance (01)

- Scan the QR Code

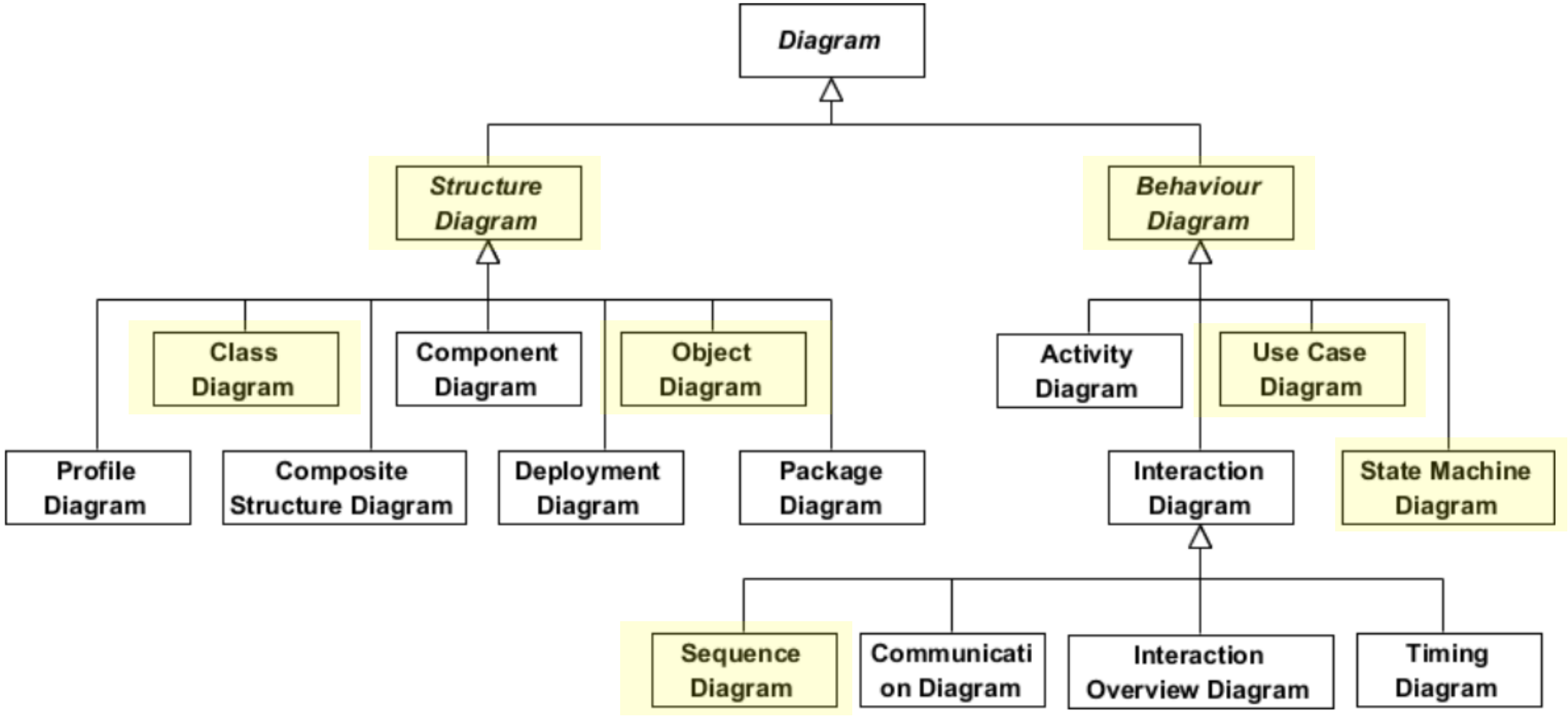
Daily Attendance (02)

- Scan the QR Code

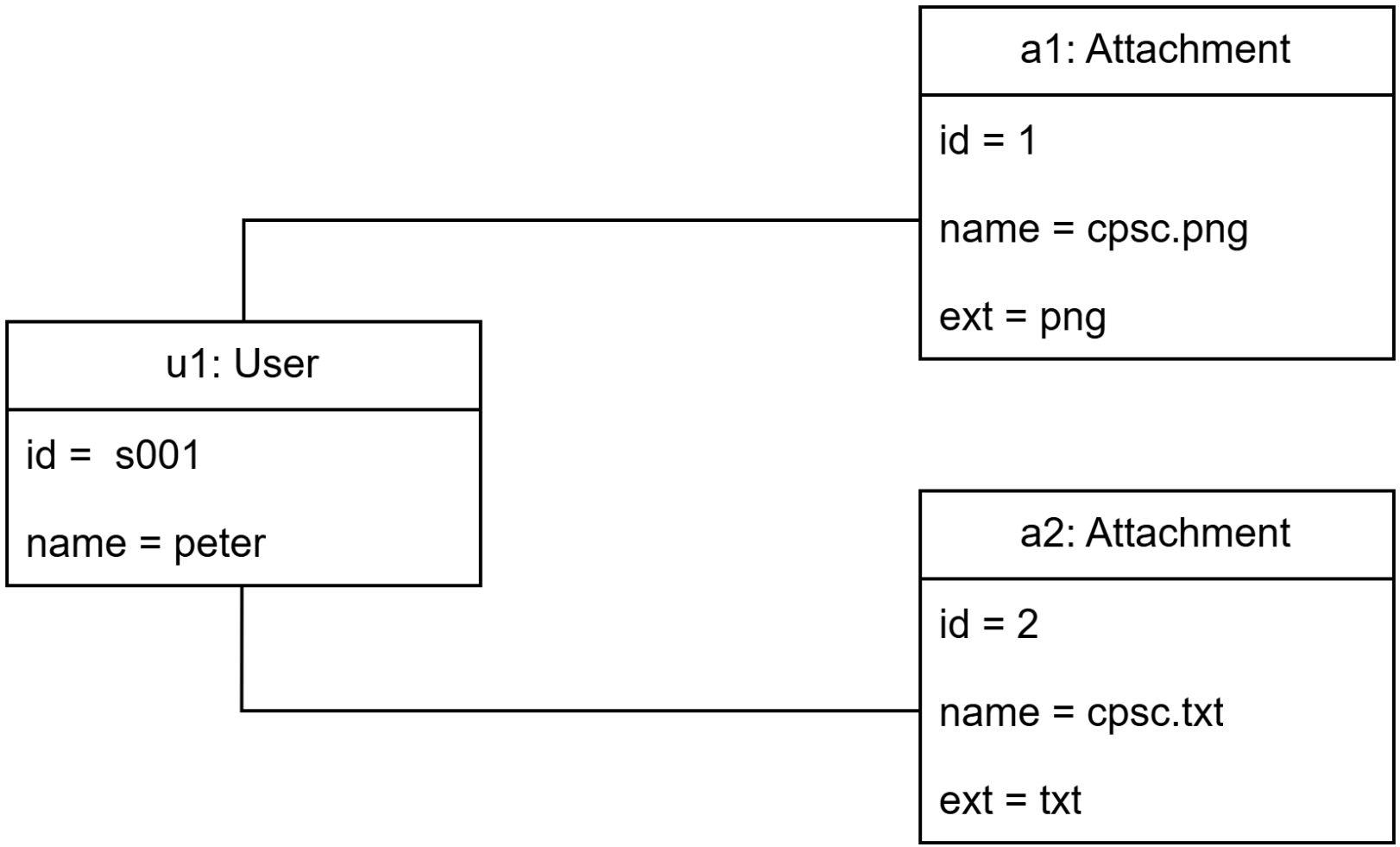
Review - Last Class

- ✓ We learned – Object Diagram
- ✓ We learned – Why do use object diagram?
- ✓ We learned – What is state machine diagram?
- ✓ We learned – State machine real life example
  - Phone dialing example
  - Traffic lights example

UML Diagrams



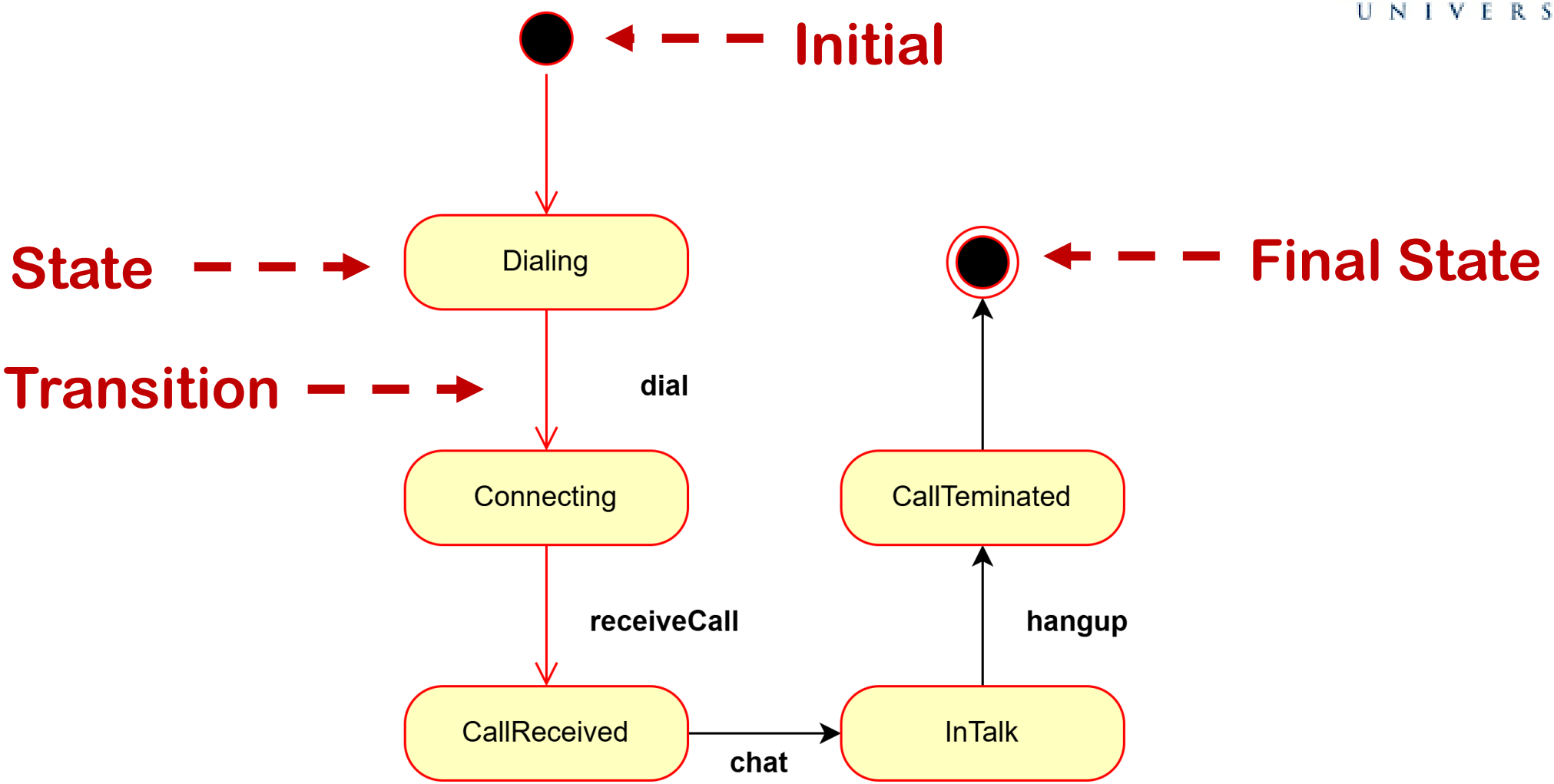
Object Diagram Example



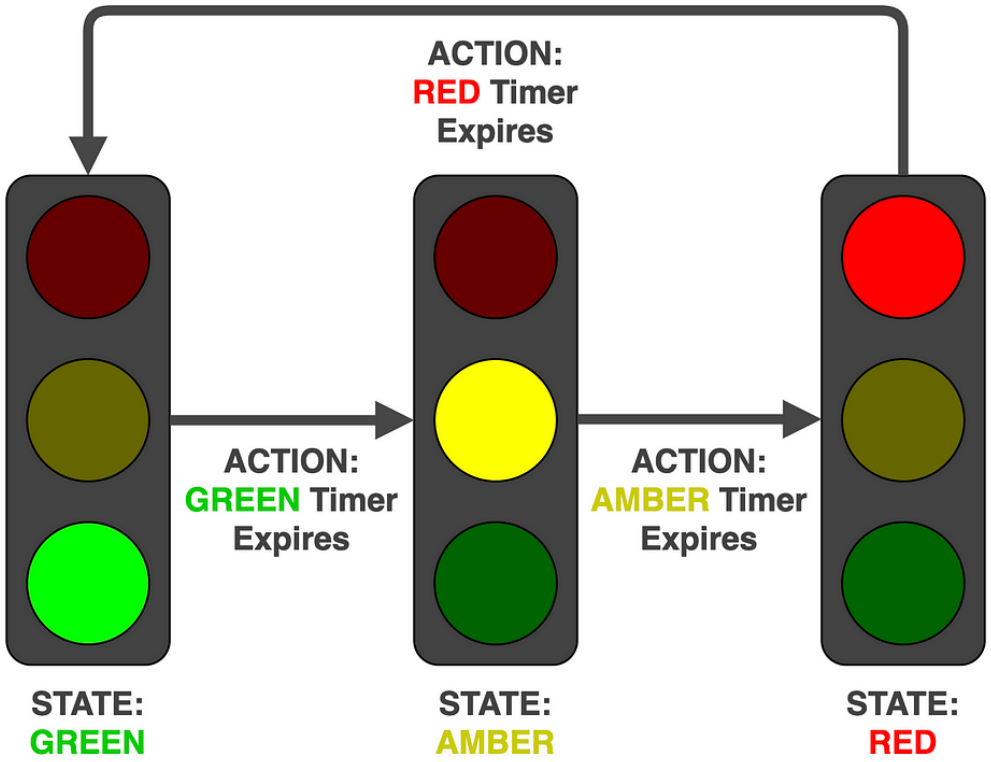
A UML Object Diagram can be seen as :

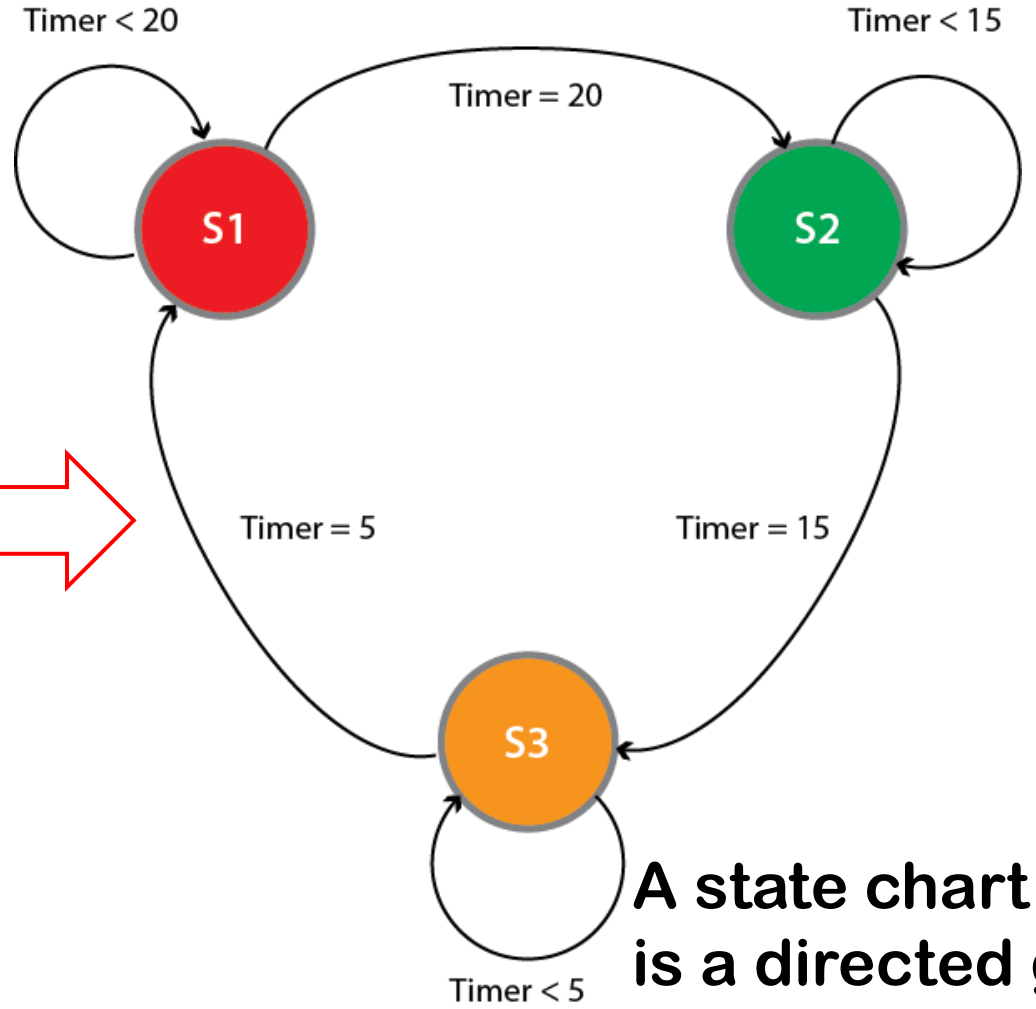
A representation of how classes are utilized at a particular state

State Machine Diagram Example



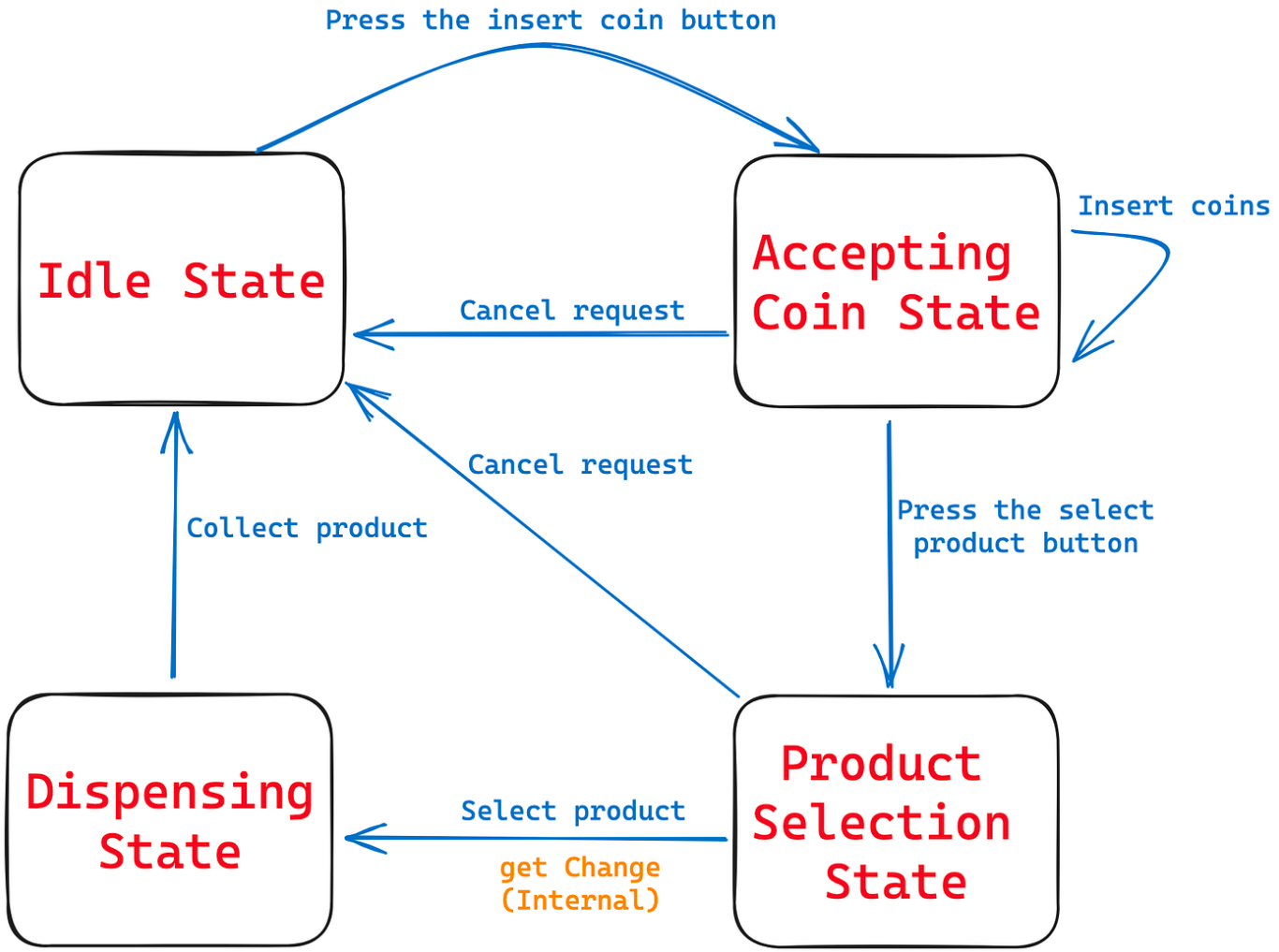
State Machine and Traffic Lights Exercise





A state chart diagram, is a directed graph

State Machine and Vending Machine Exercise



Unit Testing Introduction

Unit Testing Introduction

What is testing for?

1. The primary purpose of testing is to detect software failures, so that defects may be discovered and corrected

2. Testing also can be used to establish whether a project is complete

3. Testing is used to integrate work between different individuals or groups

Unit Testing Introduction

GONZAGA UNIVERSITY

What is testing for?

- Detects errors early
- Improves code maintainability and readability
- Makes refactoring safer

Unit Testing Introduction

GONZAGA UNIVERSITY

What is Unit Testing?

Unit Testing is a software testing technique where individual components(or unit) of a program – such as methods or functions – are tested in isolation to ensure they work correctly

Unit Testing Introduction

GONZAGA UNIVERSITY

What is Unit Testing?

- Testing individual methods in isolation
- Ensures code correctness, simplifies debugging, and prevents future bugs

Unit Testing Introduction

GONZAGA UNIVERSITY

Example:

```
function calcPayment(principal, interest, years){

}
```

Unit Testing Introduction

Example:

```
function calcPayment(principal, interest, years){

}
```

Launch the application	Check the result
Log in	Submit it
Navigate to a page	Fill out a form

Unit Testing Introduction

Unit Testing Introduction

GONZAGA UNIVERSITY

Example:

```
const result = calcPayment (10_000, 5, 5);

expect (result).toBe(188.71);
```

Unit Testing Structure

GONZAGA UNIVERSITY

Unit Test Structure: 3A

The industry area uses the AAA pattern for unit tests:

- Step 1: **Arrange**: Set up the test data
- Step 2: **Act**: Call the function/method
- Step 3: **Assert** :a truth value - Unit tests only return true or false, so do an Assert to validate the system state

Test Assertions

GONZAGA UNIVERSITY

- Assertions are the 'checks' that you may perform to determine if a test passes or fails.
- For instance(Junit5):
  - assertTrue(condition) → Passes if the condition is true.
  - assertFalse(condition)→ Passes if the condition is false.
  - assertEquals(expected, actual) → Checks if two values are equal.
  - assertSame(expected, actual) → Two objects reference the same instance?
  - assertNull(object) → Passes if the object is null.
- Generally speaking, you want ONE assertion per test.

Unit Testing vscode

Testing Java with Visual Studio Code

Edit

Testing Java in Visual Studio Code is enabled by the Test Runner for Java extension. It's a lightweight extension to run and debug Java test cases.

Overview

The extension supports the following test frameworks:

- JUnit 4 (v4.8.0+)
- JUnit 5 (v5.1.0+)
- TestNG (v6.9.13.3+)

The Test Runner for Java works with the Language Support for Java™ by Red Hat and Debugger for Java extensions to provide the following features:

- Run/Debug test cases
- Customize test configurations
- View test report
- View tests in Testing Explorer

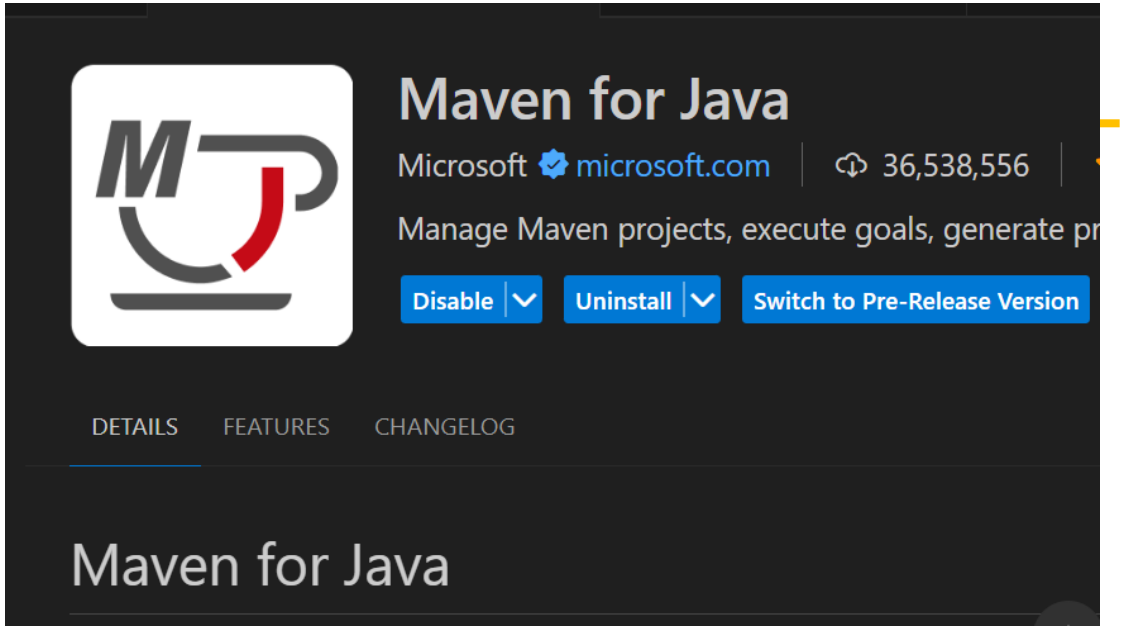
Requirements

- JDK (version 1.8 or later)
- Visual Studio Code (version 1.59.0 or later)
- Extension Pack for Java

Install the Extension Pack for Java

More details, click this link to explore:

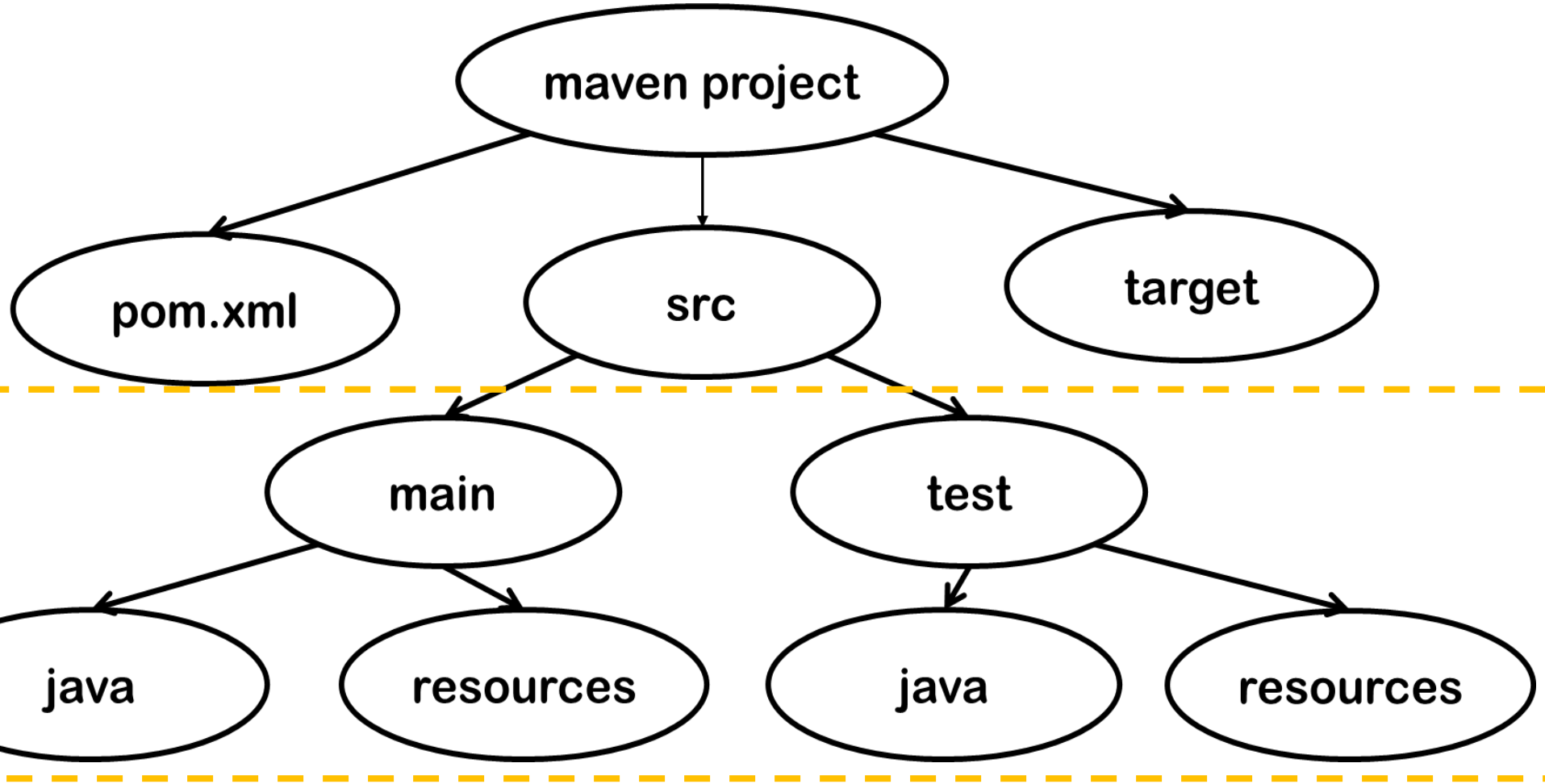
Testing Java with Visual Studio Code



Unit Testing Project Structure

GONZAGA UNIVERSITY

(pom: Project Object Model)



Unit Testing vscode

Add Junit 5 dependency in pom.xml (may need and may not need)

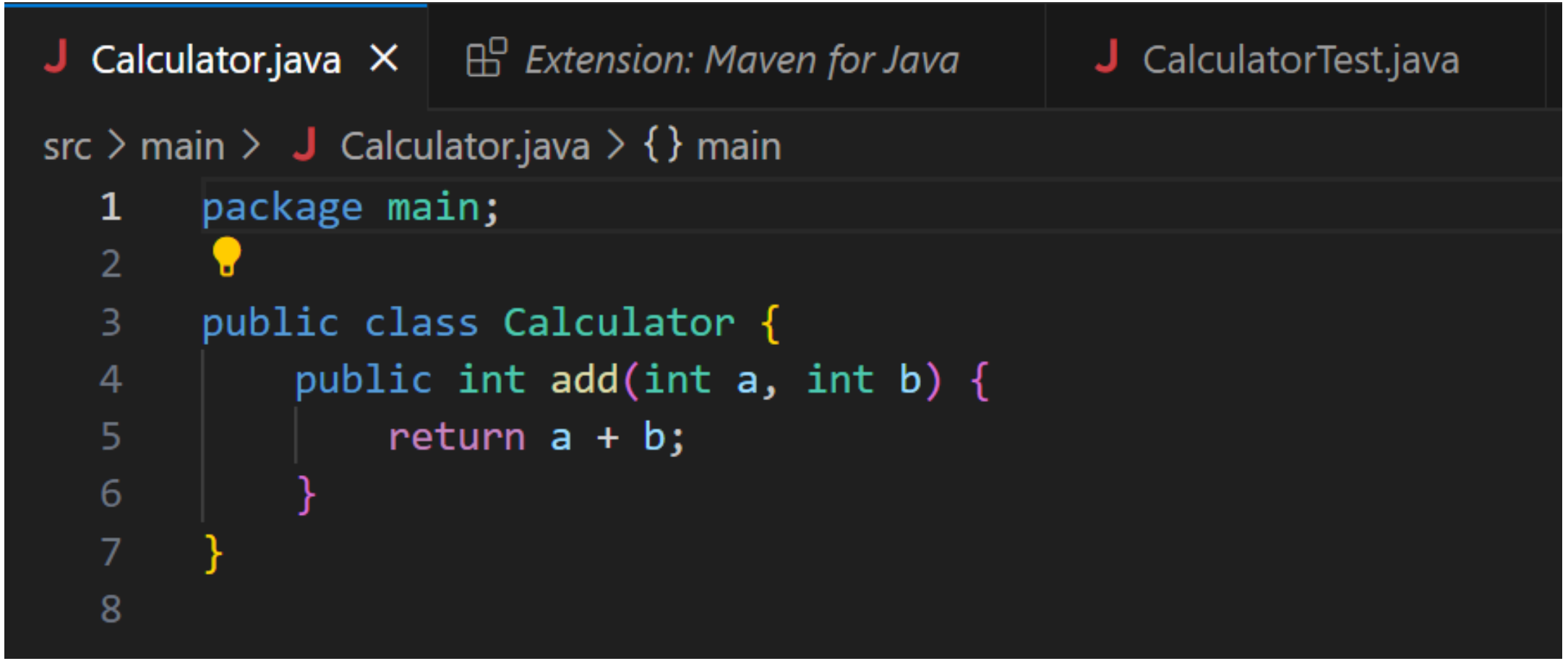
xml

```xml
<dependencies>
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-api</artifactId>
        <version>5.8.1</version>
        <scope>test</scope>
    </dependency>
</dependencies>
```

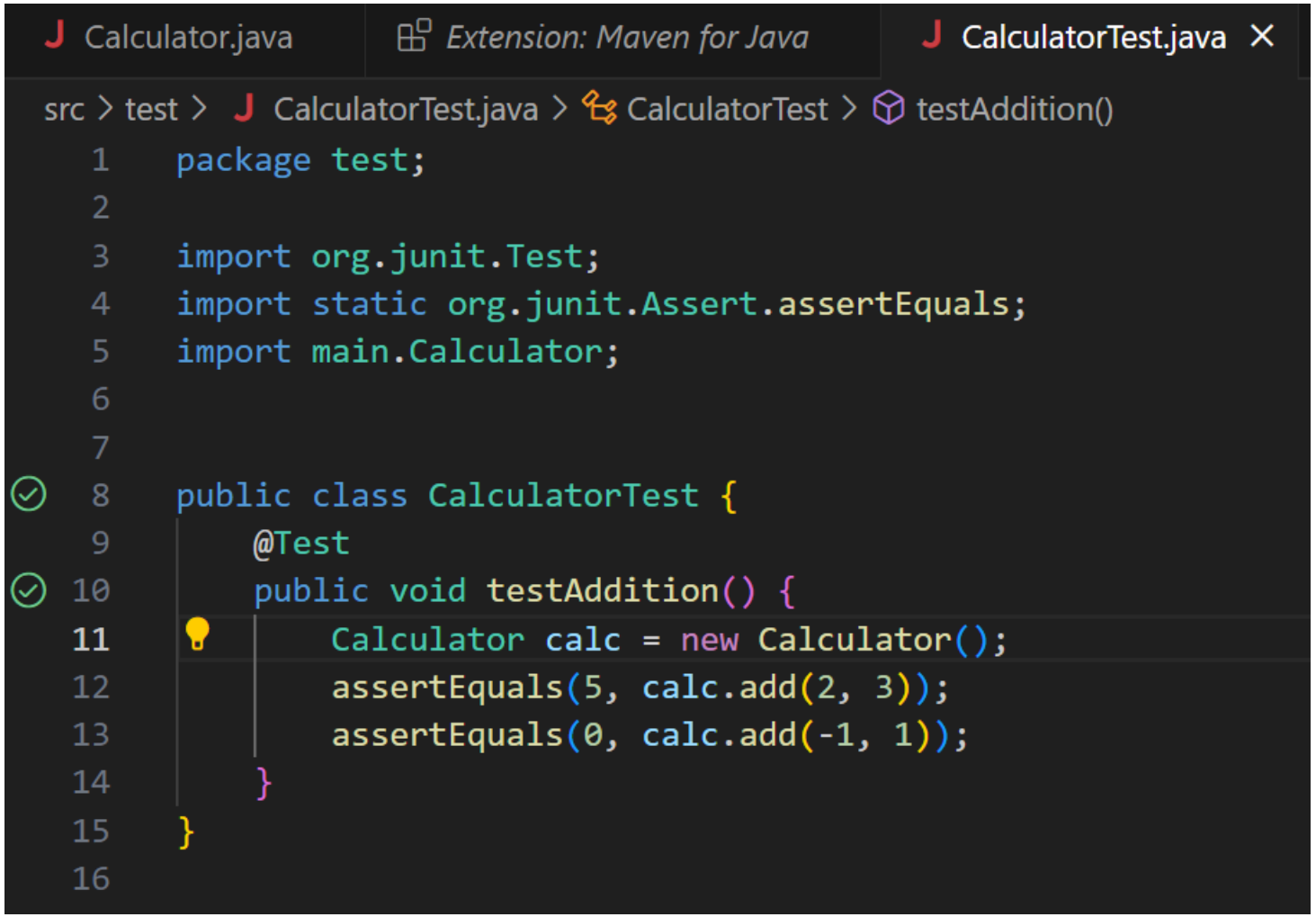
Or

```xml
<dependency>
  <groupId>org.testng</groupId>
  <artifactId>testng</artifactId>
  <version>(YOUR_TESTNG_VERSION)</version>
  <scope>test</scope>
</dependency>
```

Unit Testing vscode

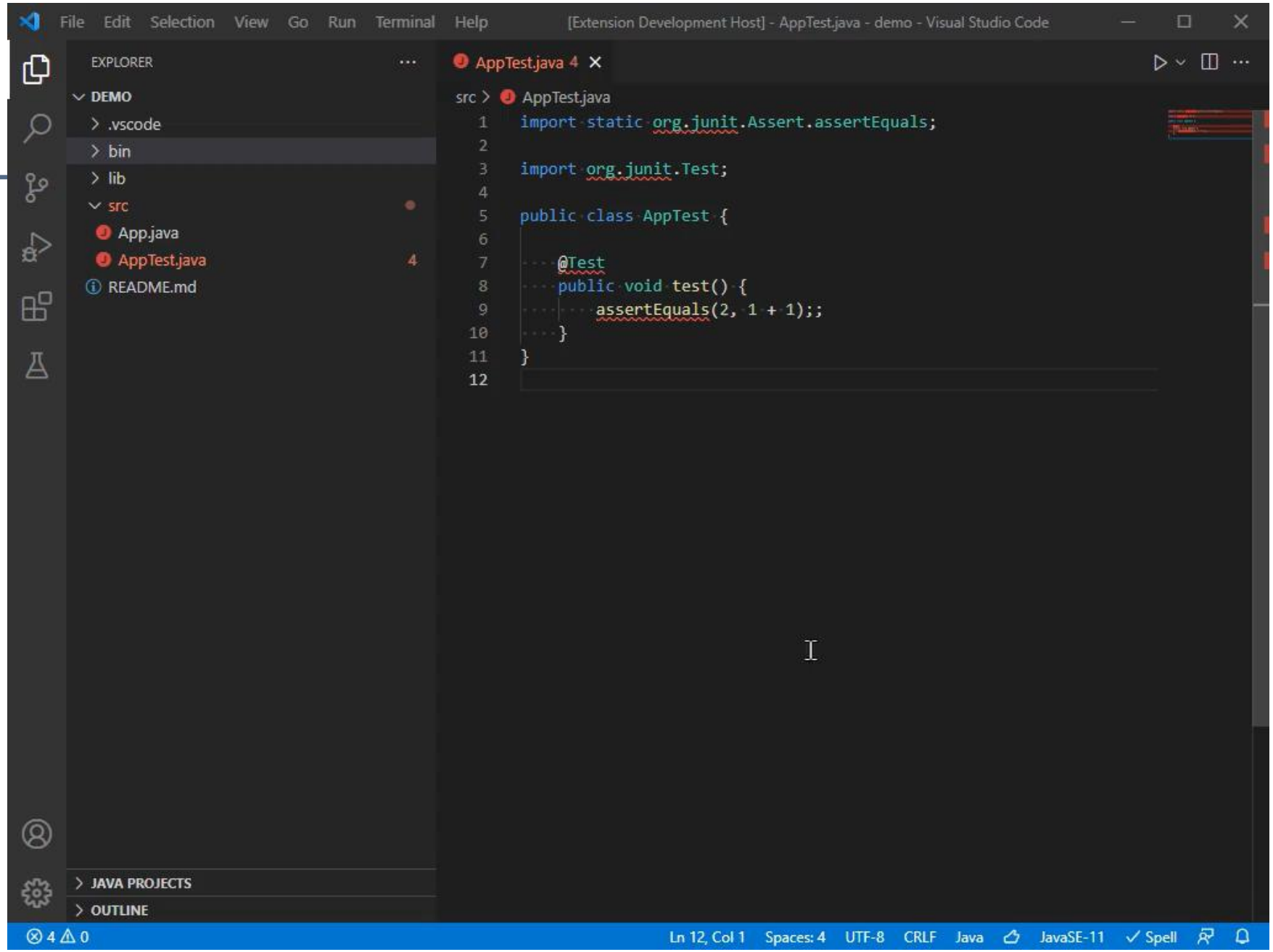


Unit Testing vscode



Unit Testing

vscode



Any Questions?