Unit Testing: Supporting Modern Software Development Methods Syllabus

UC San Diego EXTENSION

Course Number: CSE-40691
Section ID:
Quarter: Summer
Course Dates: 7 April – 9 June, 2020

Instructor Information

Name: Taylor Smith

Email: taylorsmith4546@gmail.com

Communication Policy

You may contact me either by phone (952-215-2568) or by email during my office hours. IMPORTANT: Outside of office hours, I will try to answer your emails within 24 hours of receiving them. *Please communicate with me early and often if you are experiencing challenges in completing course assignments on time.

Course Information

Course Description (Goals and Objectives)

This course will provide an overview on the benefits of unit testing, while helping students understand the value of current testing patterns. Students will learn Test Driven Development, Code Coverage, how to create tests using popular frameworks JUnit and Mockito, in addition to testing using mocks, stubs and test-spies.

The course is intended for individuals with an interest in the role that unit testing plays

within the software development life cycle, particularly those that have roles within software development teams.

Key Topics

- Learn when to use and how to write various forms of Tests; Unit, Integration, Functional
- Learn JUnit and Mockito testing frameworks
- Understand the traits of “good” and “bad” unit tests to help in writing better tests
- Use of Test Driven Development (TDD) to build a piece of software within the written tests first
- Learn how to use testing patterns with legacy code to ensure new changes will not adversely impact existing code
- Integrate testing processes within a continuous delivery model of software development

Course Materials and Textbooks

Suggested Texts:

Practical Unit Testing with JUnit and Mockito, Tomek Kaczanowski

ISBN 978-83951851-5-1

Student Learning Outcomes

By the end of this course, students will be able to:

- Grasp the role and purpose of unit tests
- Write high-quality, readable and maintainable unit tests
- Learn JUnit and Mockito
- Recognize bad unit tests and fix them
- Develop code following the TDD approach
- Use mocks, stubs and test-spies intelligently

Course Schedule

Lesson/ Week	Topics	Assignments w/due dates
1	Lecture 1: On Tests and Tools	IDE Setup Quiz 1
2	Lecture 2: Unit Testing Guidelines and Techniques	Exercise 3.11.1 Exercise 3.11.2 Exercise 3.11.5 Quiz 2
3	Lecture 3: Test Driven Development	Exercise 4.11.1 Exercise 4.11.2 Quiz 3
4	Lecture 4: Mocks, Stubs, Test Spies	Exercise 5.7.1 Read Mockito official documentation, studying its wiki page, browse Javadocs (exercise 5.7.4) Quiz 4
5	Lecture 5: Things You Should Know	Exercise 7.13.1 Exercise 7.13.2 Quiz 5
6	Lecture 6: Getting Feedback	Exercise 9.9.1 Exercise 9.9.2 Quiz 6
7	Lecture 7: Organization of Tests	Exercise 10.8.2 Read articles on BDD Quiz 7
8	Lecture 8: Maintainable Tests	Exercise 11.7.1 Quiz 8
9	Lecture 9: Test Quality	Exercise 12.8.1 Quiz 9

Grading and Assignment Information

Letter grades are based on the UC San Diego Extension Grading Scale. Your final course grade is based on the percentage of points you have earned.

Passing Grades	
A+	100%
A	93-99%
A-	90-92%
B+	87-89%
B	83-86%
B-	80-82%
C+	77-79%
C	73-76%
C-	70-72%

Non-Passing Grades	
D	60-69%
F	59% and below

Weighted Grading Criteria

Assignments (weekly projects: 8 total) = 50%

Quizzes (8 total) = 50%

TOTAL (160 points) = 100%

You can check your point total through the Course Menu in Blackboard at any time by clicking **Tools>My Grades**.

Grading Policies

This course can be taken as part of the Software Quality Assurance certificate. In order for the class to count towards your certificate it must be taken for a letter grade or as pass/no pass. Classes that are taken as NFC cannot count towards a certificate. You can change your grading option any time BEFORE the last day of class through My Extension.

Assignment Submission Policy - IMPORTANT!

1. Late assignments (anything posted or sent after the due date) will be graded -1 point for each day late unless due to a verifiable medical or family emergency.
2. Late assignments will be accepted at the discretion of the instructor and cannot be accepted more than 1 week late.
3. Assignments sent with the wrong naming convention or in the wrong format will be considered late until they are sent correctly.
4. Assignments on some weeks are not for grade and meant to benefit the student. Please do not skip these! They are just as important as the graded assignments.

Late Policy:

An assignment is considered late if it is posted or sent after the due date/time.
Late assignments will be accepted at the discretion of the instructor and cannot be accepted more than 1 week late. A couple hours or a day late is typically okay.

Assignments

There are 9 graded assignments in this course, but your final grade will be determined based solely on your top 8 assignments (The lowest homework score for each student will be dropped this quarter).

File Naming/Format:

Assignments must be submitted as a .zip file containing only your solution's .java files named following the convention LastnameFirstInitialExercise#.zip where the '#' is the Exercise number. For example, my submission for this assignment would be named TaylorSExercise1.11.zip). There should be no directories within the zip file.

Assignment Quality:

Assignments must meet all criteria as specified within their respective README files, and submitted in the correct format.

Discussion Board

Discussion Board participation is entirely optional, although highly encouraged. I will be providing articles on a weekly basis which are meant to highlight the usage of unit testing within today's larger tech companies, as well as supplementary items which will help grasp the lessons introduced within each lecture. A topic or question may be posted which complements the article to provide students a guideline for discussion.

To participate, check Course Menu>Discussion Board and look for the forum for the current week.

I will also be utilizing Slack as a way to receive quicker feedback if or when questions arise throughout the course. I will make channels dedicated for each week of class. Directions for installing Slack and adding the course chatroom will be provided in the "How-To's" section.

Quizzes

Students will be given one week following the end of the week's lecture to complete the quiz. Plan ahead and give yourself plenty of time to complete each quiz. These are based upon the weekly lesson. You may only attempt the quiz once, as these are open note and open book.

There are 9 quizzes in this course, but your final grade will be determined based solely on your top 8 quizzes (The lowest quiz for each student will be dropped this quarter).