

Module Lesson Plan : SOFTWARE TESTING AND DEBUGGING

Description
Module Facilitators
Text & Readings
Class Roster
Guest Roster
Groups
Timetable
Announcement
Chat Room
Discussion Forum
Lesson Plan
Multimedia
Project
Workbin
Library E-Reserves

Updated: 13-Feb-2014

1 2 3 4 5 6 **R** 7 8 9 10 11 12 13   



Go

Go

Week 1: 13 Jan-17 Jan

Lecture on foundations (2 hours): *Introduction to the course:* Learning outcomes and course structure, fundamentals of testing and debugging.

Hands-on hour: Basic introduction to the course project. See file [projectDescription.pdf](#)

1. Workbin [projectDescription.pdf](#)
2. Workbin [week1-lab.pdf](#)
3. Workbin [week1-lab-allpages.pdf](#)
4. Workbin [T1-fundamentals-allpages.pdf](#) - Minor additions to lecture slides, ahead of first lecture
5. Workbin [T1-fundamentals.pdf](#)

**Week 2: 20 Jan-24 Jan**

Lecture on foundations (2 hours): *Background material:* Discussion on Dependency analysis, Data-flow information, and Symbolic Execution.

Reading: Chapters 6,7 of textbook

Hands-on hour: We describe the shell that needs to be implemented (different file utilities, see project description). We also start discussion on JUnit.

Administrative issues: The class is divided into two groups to allow exchanging of code and test cases
Group 1 gets assigned grep tool and piping (extended functionality 1)
Group 2 gets assigned the text utilities (extended functionality 2)

Homework: Develop shell and start with your extended functionalities

The lecturer will be away in week 3 on Tuesday, so there will be two lectures in week 2 - Tuesday and Saturday. Thus, an additional make-up lecture will be held on Saturday 25 January 10 am at Seminar Room 1 (SR1), COM1 level 2.]

1. Workbin [JUnit-Lecture1.pdf](#)
2. Workbin [clean-code.pdf](#) - Clean Coding Guidelines
3. Workbin [eclipse-setup.pdf](#) - Eclipse installation guidelines
4. Workbin [JUnit-setup.pdf](#) - Guidelines to setup JUnit in Eclipse
5. Workbin [CodingStyle.pdf](#)
6. Workbin [cs4218-shell-project.zip](#) - CS4218 Project Interfaces (Eclipse project)
7. Workbin [T2-background.pdf](#) - Minor additions, additional slides for tomorrow's lecture.
8. Workbin [T2-background-allpages.pdf](#)

**Week 3: 27 Jan-31 Jan**

Lecture on foundations (2 hours): Finish discussion on background materials.

[The lecturer will be away in this week on Tuesday. For this reason, a make-up lecture will be held on previous Saturday 25 January 10 am at Seminar Room 1 (SR1), COM1 level 2. Webcast of this lecture is available. Lab hour on Wed proceeds as scheduled.]

Reading: Chapter 6,7 of textbook.

Hands-on hour: Continue work on shell and extended functionality. Setup Eclipse and JUnit environment for your project according to the following guidelines (1) and (2). Finish discussion on JUnit.

1. [Workbin JUnit-Lecture2.pdf](#)
2. [Workbin T2-background.pdf](#) - Minor additions, additional slides for tomorrow's lecture.
3. [Workbin T2-background-allpages.pdf](#)
4. [Multimedia Makeup Lecture - Discussion on Dependency analysis, Data-flow information, and Symbolic Execution.](#)



Week 4: 03 Feb-07 Feb

Lecture on foundations (2 hours): Functional or specification based testing

Reading: Chapter 10,14 of textbook

Hands-on hour: Continuation of JUnit, and its usage on the code. Students are encouraged to start using Junit in their projects, as they are developing the code.
Underline the differences between requirements driven development and test driven development.

Homework: Finish development of shell and your extended functionality.

Project Milestone: Submit the test cases for the shell and extended functionality before week 5

1. [Workbin JUnit-Lecture2.pdf](#)
2. [Workbin Week4-lab.pdf](#)
3. [Workbin T3-functionaltesting-allpages.pdf](#) - Minor additions ahead of tomorrow's lecture
4. [Workbin T3-functionaltesting.ppt](#)



Week 5: 10 Feb-14 Feb

Lecture on foundations (2 hours): Combinatorial testing

Reading: chapter 11 of textbook

Hands-on Hour: Test Driven Development. Swap the test cases between the groups.

Homework: Develop the other extended functionality based on the newly acquired test cases

1. [Library : E-Reserves Combinatorial testing.PDF](#)
2. [Workbin cs4218-TDD-example.zip](#)
3. [Workbin TDDWeek.pdf](#)
4. [Workbin T4-combinatorialtesting.pdf](#)
5. [Workbin T4-combinatorialtesting-allpages.pdf](#)



Week 6: 17 Feb-21 Feb

Lecture on foundations (2 hours): Structural or White-box testing based on control and data flow coverage

Reading: Chapter 12,13 of textbook

Hands-on hour: Complete the code development in the project.

Homework: Write integration tests. Make sure that integration test invokes all the utilities in combinations with at least one other utility.

1. [Workbin IntegrationTesting.pdf](#) - Week 6 - Hands-on Hour
2. [Workbin T5-whitebox-testing.pdf](#)
3. [Workbin T5-whitebox-testing-allpages.pdf](#)



Week Recess: 22 Feb-02 Mar

[Answers to the non-graded quiz questions which were posted in week 5, are also posted.]

1. [Workbin CS4218Quiz.pdf](#) - Practice questions for you to try. These questions are not graded, and they are not an indicator...
2. [Workbin Quiz - answers.pdf](#) - Answers to the non-graded quiz questions posted in week 5.



Week 7: 03 Mar-07 Mar

LECTURE on foundations (2 hours): Testing of O-o programs (*the materials in this lecture are for discussion only, and will not be tested in mid-term or final examination*).

Reading: Chapter 15 of textbook

Hands-on hour: Discuss code coverage tools. Determine the statement, branch, and MC/DC coverage of your code.

Homework: Add relevant test cases (if needed).

-
1. [Workbin Coverage.pdf](#) - Week 7 - Hands-on Hour
 2. [Workbin org.apache.commons.collections.zip](#) - Week 7 - Hands-on Hour Additional Material
 3. [Workbin org.eclEmma.samples.simplemath.zip](#) - Week 7 - Hands-on Hour Additional Material
 4. [Workbin T6-oo-test.pdf](#)
 5. [Workbin T6-oo-test-allpages.pdf](#)

**Week 8: 10 Mar-14 Mar**

Lecture on foundations (2 hours) (First hour) Midterm + (Second hour) Mutation testing, test execution

Reading: Chapter 16, 17 of textbook

Additional Reading: A practical approach to programming with assertions, David Rosenblum, ICSE 1992, journal paper in TSE January 1995.

Hands-on hour: left flexible this week, can discuss course materials outside course project specifically for this week.

Project Milestone: *Submit all code, test cases and coverage reports by this week Friday March 14 11:59 pm, at the folder Workbin -> Project-Milestone2*

-
1. [Workbin T7-Mutation-testing.pdf](#) - Added two examples (slides 9,10) rest of slides are as before
 2. [Workbin T7-MutationTesting-allpages.pdf](#) - Added two examples (slides 9,10) rest of slides are as before

**Week 9: 17 Mar-21 Mar**

Lecture on foundations (2 hours): Understanding failures, Writing of QA reports, Reproducing Program Errors for Debugging, Tracking value origins - dependencies, slicing

Reading: Chapter 4,7 of supplementary text & Chapter 23, 24 of textbook.

Hands-on hour: Hackathon. Hand-over of one group's code to another. Limited interaction in explanation across teams to simulate industrial practice. Generate test cases that fail due to bugs in others code (if any). Determination of testing strategies to test other's code.

-
1. [Workbin T8-Debug.pdf](#)
 2. [Workbin T8-Debug-allpages.pdf](#)

**Week 10: 24 Mar-28 Mar**

Lecture on foundations (2 hours): Fault Localization: Trace comparison, Anomaly analysis [finish discussions on debugging, and distinguish debugging from model checking / formal verification] *We also start the discussions on model-based testing.*

Reading: Chapter 11, 12 of supplementary text [additional online materials will be given to students for reading]

Hands-on hour: Quality Assurance. Comparison of testing strategies, measures of confidence.

Homework: Write a quality assurance report.

-
1. [Workbin T8-Debug.pdf](#)
 2. [Workbin T8-Debug-allpages.pdf](#)
 3. [Workbin T9-ModelbasedTesting.pdf](#)
 4. [Workbin T9-ModelbasedTesting-allpages.pdf](#)



Week 11: 31 Mar-04 Apr

Lecture on foundations (2 hours): We finish our discussion on model-based testing, and discuss how it ties up with formal verification methods such as model checking.

Hands-on Hour: Debugging strategies for the failed test cases that were found earlier.

Homework: Debug and fix your code such that all test cases pass.

Project Milestone: *Submit all test cases with bug reports, from the Hackathon week (specifically, the failing ones).*

1. [Workbin T9-ModelbasedTesting.pdf](#)
2. [Workbin T9-ModelbasedTesting-allpages.pdf](#)

**Week 12: 07 Apr-11 Apr**

Lecture on foundations (2 hours): Advanced research topics on software testing - symbolic execution based testing, test suite augmentation and the need for test suite evolution. Students will be given advanced research papers, and they will critique selected papers. Papers to be discussed are as follows.

DART: Directed automated Random testing, PLDI 2005 paper by Godefroid, Klarlund and Sen, see http://research.microsoft.com/en-us/um/people/pg/public_psfiles/pldi2005.pdf

KLEE: Unassisted and automatic generation of high coverage tests for complex systems programs, OSDI 2008 papers, see <http://www.stanford.edu/~engler/klee-osdi-2008.pdf>

Test generation to expose changes in evolving programs, ASE 2010 paper by Qi, Roychoudhury and Liang, see <http://www.comp.nus.edu.sg/~abhik/pdf/ase10.pdf>

Hands-on hour: Finish debugging, and generate fixes for failed tests.

**Week 13: 14 Apr-18 Apr**

Lecture on foundations (2 hours): Advanced /research topics in software debugging and fixing: Symbolic Execution based Debugging, Statistical fault localization, Automated repair strategies (not all may be covered)

Reading: Recent research articles will be given to the students, and they will write a critique at least one of the papers. Sample papers to be read: see the papers

"Yesterday, my program worked. Today it does not. Why", by Zeller, see <http://www.st.cs.uni-saarland.de/publications/files/zeller-esec-1999.pdf>

"A few billion lines of code later. Using static analysis to find bugs in the real world", CACM, <http://www.stanford.edu/~engler/BLOC-coverity.pdf>, also see <http://www.stanford.edu/~engler/mc-osdi.pdf>

"DARWIN: An approach for debugging evolving programs" by Qi, Roychoudhury, Liang and Vaswani, at <http://www.comp.nus.edu.sg/~abhik/pdf/fse09.pdf>

Hands-on Hour: Finalization and validation of fixes, by re-running tests. Discussion of how test selection was done during the re-running.

**Week Reading: 19 Apr-25 Apr**

Project Milestone: Finalization and Submission of Project Report, Initial Code, Tests, Final Code, QA report plus any other artifacts (such as debugging logs).

