



Philadelphia University
Faculty of Information Technology
Department of Software Engineering
Second Semester, 2015/2016

<u>Course Syllabus</u>	
Course Title: Software Testing	Course code: 0721430 ;721386;761467
Course Level: 3	Course prerequisite (s) and/or corequisite (s): 721385
Lecture Time: 13:10 -14:00	Credit hours: 3

Academic Staff Specifics

Name	Rank	Office No.	Office Hrs.	E-mail Address
Dr. Mohammad Taye	Assistant Professor	309	-S-T-T 11:00 -12:00 12:00-13:00	mtaye@philadelphia.edu.jo

Course description:

This course will address topics in the verification and validation (V&V) of software. *Verification* addresses issues related to whether the system is correct (with respect to some specification), *validation* addresses the question whether the right system was built. An in depth study of verification and validation strategies and techniques as they apply to the development of quality software. Topics include test planning and management, testing tools, technical reviews, formal methods and the economics of software testing. The relationship of testing to other quality assurance activities as well as the integration of verification and validation into the overall software development process are also discussed.

Course objectives:

- Understand the concepts and theory related to software testing.
- Understand different testing techniques used in designing test plans, developing test suites, and evaluating test suite coverage
- Understand the relationship between black-box and white-box testing and know how to apply as appropriate.
- Learn to use automated testing tools in order to measure code coverage.
- Understand how software developers can integrate a testing framework into code development in order to incrementally develop and test code.

Course/ module components

- **Books (title , author (s), publisher, year of publication)**

1. Software Verification and Validation for Practitioners and Managers, by Steven R. Raktitin, ed. Artech House, 2nd Edition, ISBN 1-58053-296-9, 2001,

- **Support material (s) (Slides, Softcopy textbook, etc).**
- **Study guide (s) (if applicable)**
- **Homework and laboratory guide (s) if (applicable).**

Teaching methods:

Duration: 15 weeks, 45 hours in total. Based on Lectures, Tutorials, Problem solving.

Learning outcomes:

- Knowledge and understanding
- Understand the role and importance of software quality assurance in software verification
- Understand the role and importance of software inspection in software verification
 - Understand the concepts and theory related to software testing.
 - Understand different testing techniques used in designing test plans, developing test suites, and evaluating test suite coverage
 - Understand the relationship between black-box and white-box testing and know how to apply as appropriate.
 - Learn to use automated testing tools in order to measure code coverage.

- Cognitive skills (thinking and analysis).
 - Be able to perform peer reviews and inspections for defect prevention.
 - Be able to design test cases based on different testing methods.
 - Be able to process testing in all phases of the software development lifecycle
 - Be able to apply and manage V&V to software engineering to improve the software development process, software quality and reliability.
- Communication skills (personal and academic).
 - Be able to communicate with other member of V&V team in order to conduct an efficient V&V process.
 - Be able to communicate with other member of software engineering project in order to transmit information about founded bugs.
- Practical and subject specific skills (Transferable Skills).
 - Be able to follow the process of V&V
 - Be able to prepare forms and reports of V&V via a standard

Assessment instruments

- Application of software V&V process on a full software project with presentation.
- Quizzes.
- Assignment
- Final examination: 40 marks

<u>Allocation of Marks</u>	
Assessment Instruments	Mark
First examination	20
Second examination	20
Final examination: 50 marks	40
Quizzes	20
Total	100

Documentation and academic honesty

- Documentation style (with illustrative examples)

Documentation style of slides, exam, tutorial and assignment presentation are available.

Academic honesty is based on the principle that one’s work is one’s own. We encourage all students to accept responsibility for taking academic honesty seriously by being informed, by contributing to a climate in which honesty is valued, and by considering responsible ways to discourage dishonesty in the work of others. Students, faculty, administrators and staff should not condone or tolerate cheating, plagiarism, or falsification, since such activity negatively affects members of the academic community. Plagiarism is the presentation of all or a portion of someone else’s work, as one’s own, without properly citing/documenting the work. **Plagiarism is unacceptable and will result in a failing grade in the course.**

Course/module academic calendar

week	Basic and support material to be covered	Homework/reports and their due dates
(1)	Introduction to Software Testing	
(2)	Software testing overview	
(3)	Test data & test cases & Testing Levels	
(4),(5)	Testing Strategies	Assignment Date Set Tutorial 1
(6) First examination	Software quality metric	
(7)	Testing Tools, Assessing and improving the validation process	
(8)	Functional & Structural testing	
(9)	Integration and system testing	Tutorial 2, Tutorial 3
(10), (11) Second examination	Equivalence Class Testing	Tutorial 4

(12)	Performance and Load Testing	Tutorial 5
(13)	Web based Testing	
(14)	Regression Testing	
(15)	Improving the development process	
(16)	Final exam	

Expected workload:

On average students need to spend 2 hours of study and preparation for each 50-minute lecture/tutorial.

Attendance policy:

Absence from lectures and/or tutorials shall not exceed 15%. Students who exceed the 15% limit without a medical or emergency excuse acceptable to and approved by the Dean of the relevant college/faculty shall not be allowed to take the final examination and shall receive a mark of zero for the course. If the excuse is approved by the Dean, the student shall be considered to have withdrawn from the course.

Module references

Books

Title [Managing the Testing Process](#)

Author Rex Black

Publisher John Wiley & Sons

Publication Date August 2009

ISBN 0470404159

Software Testing: Principles and Practices by Srinivasan D and Gopalswamy R, PearsonEd, 2006

Journals

None at this level

Websites

<http://www.cs.drexel.edu/~spiros/teaching/SE320/index.html>