

COMP 5222
Software Testing and Quality Assurance
Teaching Plan
2012-2013, Term 1

Subject leader: Dr. Hareton Leung



csheung@comp.polyu.edu.hk



PQ817



<http://www4.comp.polyu.edu.hk/~csheung/>



2766 7252

Objectives:

1. Understand issues, techniques, and metrics for quality assurance.
2. Understand issues, techniques, and tools for software testing.
3. Distinguish different types of testing techniques (black box and white box) and their relative strengths and weaknesses.
4. Gain the ability to apply testing to complex applications, select among candidate testing techniques, and use testing techniques in combination.
5. Apply the learned testing and quality assurance techniques to improve software quality.

Pre-Knowledge: Knowledge of software development process at a level equivalent to a typical undergraduate software engineering subject. Some working experience in software development is desirable.

Format:

The subject will be structured as lectures and tutorials, involving class discussions and external reading. You are expected to attend all classes and come to class having read the assigned material and prepared to discuss it.

Download the lecture notes from Blackboard 2-3 days before the lecture date.



Examination	55%
Continuous Assessment	45%
Assignment 1	15%
Assignment 2	15%
Project	15%
Total	100%

- | |
|---|
| ⇒ Assignments and project <u>are due in class</u> . |
| ⇒ To be fair, late work will be accepted with a 5% penalty per day . |
| ⇒ Your work will be evaluated on correctness, clarity, and quality. |

⇒ Exam is closed book.

No copying of work will be tolerated. Students will receive zero marks for any assessment work that is very similar to work submitted by another student.

Your assignments and project may be submitted to www.turnitin.com for checking their originality.

References:

1. Black, R., Pragmatic Software Testing: Becoming An Effective and Efficient Test Professional, Wiley, 2007.
2. IEEE Std. 829, 1008, 1012, 1059, 1028, 1044
3. Whittaker, J. A., How to break software, Addison-Wesley, 2003.
4. How Google do their testing? <http://googletesting.blogspot.com/>
5. Software Testing and Quality Bookshelf
<http://www.soft.com/Institute/QualitySource/name.list.html>
6. Software QA Testing and Test Tool Resources, <http://www.aptest.com/resources.html>

Tentative Schedule:

<i>Week</i>	<i>Topic</i>	<i>Deliverable</i>
1 (21/9)	Software Quality Assurance (Quality model, Quality factors, Cost of quality, QC, QA)	
2 (28/9)	Testing Fundamentals (Understanding defects, Testing concepts, Levels of testing, Test process, Test plan)	
3 (5/10)	Testing Fundamentals	
4 (12/10)	<i>No Lecture</i>	
5 (19/10)	Code-based Techniques (Control flow and data flow testing, Mutation testing, Domain testing, Error-oriented testing)	
6 (26/10)	Code-based Techniques	
7 (2/11)	Specification-based Techniques (Equivalence partitioning, Boundary value testing, State machine testing, Orthogonal array, Cause-effect graph, Program verification)	<i>Assignment 1</i>
8 (9/11)	Specification-based Techniques	
9 (16/11)	Inspection (Cost and benefits, Team and roles, Process)	
10 (23/11)	Test Tools (Review, Test Planning, Test Design, Execution, Support, Automation case study)	<i>Assignment 2</i>
11 (30/11)	Measuring Software Quality (Metrics for test product, process and resources, ROI, Estimate defects, Testing Maturity Model)	
12 (7/12)	Measuring Software Quality TDD (Test last, Test first, Steps, Benefits)	
13 (14/12)	Project presentation	<i>Project due</i>
14 (21/12)	<i>No Lecture (Winter Solstice)</i>	