Introduction to Software Engineering

SFEN310B – Introduction to Software Engineering

Instructor: Mahindrapersad Koon
Class Schedule: Tuesdays/Wednesdays 8:00 a.m. – 10:00 a.m.
Location: Classroom 6
Lab time/location: n/a
Office Hours: 10:00 a.m. –12:00 noon Monday or preferably by appointment
E-mail: mahindrapersad.koon@utt.edu.tt
Course URL: www.u.tt/ict

Course Description

This course provides an introduction to software engineering including the fundamental activities of specification, design, implementation, testing, and maintenance with a focus on UML modeling techniques.
Exercises and demonstrations emphasize good development practices. Students are introduced to basic Object-Oriented concepts including development examples in Java. A major component of the course is a semester long, group project where students apply the software engineering fundamentals to a medium-scale development project.

Assessment

Assignments 20 % on a weekly basis as part of the semester project-individual and group.
Project 40 %
Midterm Exam 15 % (50 minutes in class)
Final Exam 25 % (cumulative)
No late assignments will be accepted.

Reference Material

Roger Pressman, *Software Engineering: A Practitioner's Approach 6th ed.,* ISBN 0-07-285318-1, 2005. QA76.758P75 2005;
Martin Fowler, *UML Distilled* 3rd ed, ISBN 0-321-19368-7 2004 QA76.9.O35F69 2004
Stephen R. Schach, *Object-Oriented Classical Software Engineering 7th ed.* ISBN-10 0-07-319126-4, QA76.78.534 2007
Perdita Stevens, *Using UML with Objects and Components* ISBN 0-201-64860-1

Course Schedule

Date		Topics and Reading
Week 1	January 13 (T) Introduction January 14 (W) What is Software Engineering	General Discussion, expectations etc. Motivation, benefits, profession (Ch. 1)
Week 2	January 20 (T) What is Software Engineering January 21 (W) Requirements Specification phase	Ian Sommerville Ch 1 Waterfall Lifecycle Model
Week 3	January 27 (T) Introduction to OO Concepts January 28 (W) Specification	Classes, Objects, Inheritance Generalization, Specialization, Association
Week 4	February 3 (T) Life Cycle Models February 4 (W) Life Cycle Models	Prototyping /Rapid Application Development Models Incremental/Spiral Development Models
Week 5	February 10 (T), and 11 (W) Unified Modeling Language(UML)	Detailed Class Diagrams, Aggregation/Composition Use Case Modeling, Actors etc
Week 6	February 17 (T) UML February 18 (W) Introduction to Objects in Java	Sequence Diagrams Writing Classes in Java
Week 7	February 24 (T) Carnival February 25 (W) ASH Wednesday	NO CLASS To Be Advised
Week 8	March 3 (T) Classes and Objects in Java March 4 (W) Mid-Term Examination	Programming (Laptops in the Classroom) Writing Classes in Java cont'd
Week 9	March 10 (T) Classes and Objects in Java March 11 (W) Reuse and Reusability	Programming (Laptops in the Classroom) Patterns, inheritance, problems in reuse
Week 10	March 17 (T) Modularity March 18 (W) Software Testing Strategies	Cohesion and Coupling Pressman, pg 303-321 Pressman, Ch 13
Week 11	March 24 (T) Project Management Concepts March 25 (W) Software Project Estimation	Teamwork, Planning, Gantt Charts etc. Pressman, Ch 21 CoCoMo and FP estimation, Pressman Ch 23
Week 12	April 6 (T) Risk Management Concepts April 7 (W) Other Issues in SE	Pressman, Ch 25 Agile Development, Extreme Programming, Synchronize and Stabilize. Algorithm Analysis and Design-Big O