**Graduate Program in Software** 

**SEIS 610: Software Engineering** 

**Syllabus** 

# Syllabus

# Graduate Program in Software SEIS 610: Software Engineering

**Instructor** 

Name: Chih Lai, Ph.D.

E-mail: clai@stthomas.edu

(please include "610" in the subject field of your e-mail)

WWW: <a href="http://personal1.stthomas.edu/clai/">http://personal1.stthomas.edu/clai/</a>

Voice: 651-962-5573 Fax: 651-962-5543

Mailing stop: Mail #OSS301

University of St. Thomas

2115 Summit Avenue, St. Paul, MN 55105-1079

Office: OSS 308

Office Hours: 4:00 - 5:00 PM Wednesday

Also by prior appointment.

#### **Class Rooms / Hours**

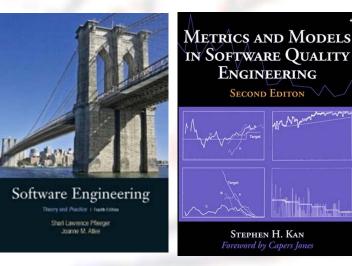
OSS 313, 5:45 PM – 9:00 PM Wednesday

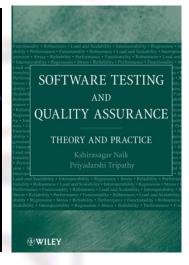
### **Textbooks**

- B1. Software Engineering: Theory and Practice, by Shari Lawrence Pfleeger, Prentice Hall 2009.
- B2. Metrics and Models in Software Quality Engineering, Stephen H. Kan, Addison, 2003.

B3. Software Testing and Quality Assurance, by Kshirasagar Naik, and Priyadarshi

Tripathy, Wiley, 2008

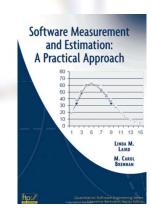


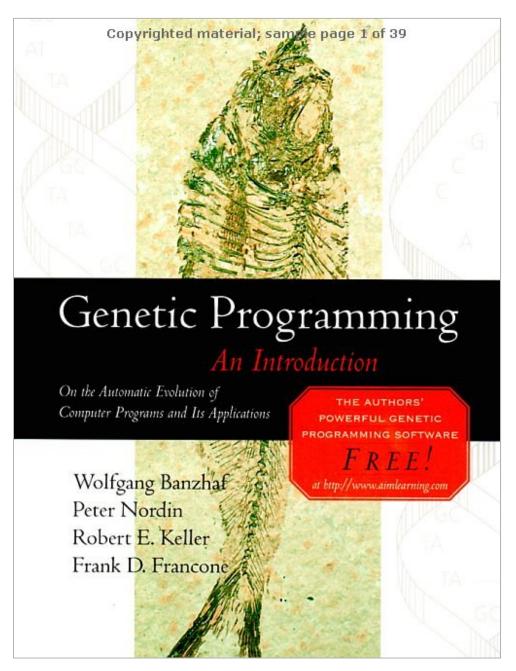


### **Optional Textbooks**

Software Measurement and Estimation, by Linda M. Laird, and M. Carol Brennan, IEEE/Wiley, 2006.

Genetic Programming: An Introduction On the Automatic Evolution of Computer Programs and Its Application, Morgan Kaufuman, 1998





# Course Description

### **Course Description**

To introduce fundamental concepts, terminology, and methodologies involved in current and (possible) future software engineering process. We will use current software engineering tools to build a possible future software engineering environment— *Genetic Programming* (GP) environment.

By building this GP project, we can

- (1) simulate current software engineering processes,
- (2) **explore/measure/**predict difficulties of current software engineering techniques,
- (3) introduce fundamental data structures and algorithms for building software

"The real world is messy. Mostly what it takes to get a (software) product to market is the ability to organize your work, collaborate, and persevere."

— <u>http://www.ibm.com/developerworks/rational/library/dec05/pollice/index.html</u>

#### **Prerequisite**

Some programming experiences / SEIS 601 Foundation of Software Development.

# Tentative Class Schedule– Session 1~7

	Date	Topics		
1	9/4	* Genetic Programming introduction  * Software engineering introduction		
2	9 / 11	* Software engineering overview, software development phases  * Analysis, data modeling, object-oriented modeling		
3	9 / 18	* Analysis, functional modeling—data flow diagram  * Analysis, behavior modeling— Finite State Machines (FSM)		
4	9 / 29	* Modeling using Petri net, Petri-net for distributed systems and locking  * Timed Petri-net, (time permitted)  * Design—coupling, cohesion, reducing design complexity		
5	10 / 2	* Fault tolerance design—critical state method  * Fault-tree analysis, error control coding		
6	10/9	* Data structures—design and analysis of algorithms, stacks, trees, tree-traversal  * Data structures and algorithms for genetic programming  * B / B+ tree, high-dimensional search (time permitted)		
7	10 / 16	* Software configuration management (SCM)  * Documenting/Analyzing SW Architecture  * Project plan due		

# Tentative Class Schedule– Session 8~14

	Date	Topics		
<u>8</u>	10/23	* Mid-term exam  * Group project discussion		
9	10 / 30	* Software measurements— LOC, functional points, fan-in, fan-out  * Metrics: Cyclomatic, Halstead's, info. flow, maintainability, object-oriented metrics		
10	11 / 6	* Call graph, call size, call depth  * Software quality— availability, maintainability, reliability  * Defect removal efficiency, defect benchmark data		
11	11 / 13	* Software testing—black/white box testing, verification/validation  * Unit testing & coverage (statement, branch, condition, path), loop test, control graph		
12	11 / 20	* Functional test— equivalence class partition, boundary value analysis, decision table  * System testing— regression testing, alpha/beta testing  * Mutation test, fault-seeding, automatic testing with genetic algorithm  * Data flow testing, finite state machine testing		
13	12/4	* Project presentation (do NOT discuss programming details)  * Notes prepared by teams		
<u>14</u>	12/11	* Final Exam  * Final Project due		

# Course Project – Genetic Programming Environment

### **Course Project**

You will conduct a small software project in a team of <u>5-6</u> people (*how do you break down to teams*??).

The instructor will offer general guidelines, but not specific coding details to each team.

The project plan (due around midterm) must include the following sections:

- (1) Problem Description,
- (2) Requirements Analysis and Preliminary System Design,
- (3) Weekly SCM files and folders for traceability (what where how who time), in plain text files
- (4) Work Plan (breakdown among members, milestones, estimated schedules, hours).

Each team must demonstrate its project and submit a final report with the following sections:

(1) All sections refined from the project plan,

(2) Final System Design,

(3) System Testing Plan and Test Results,

(4) Lessons Learned,

(5) Project Summary,

- (6) Weekly SCM files / folders,
- (7) Visual Materials (document printout, slides),
- (8) A (short) User Manual,

- (9) Post-project analyses
- (10) All the project related materials (documents, slides, programs, and executable files).

# Exams and Grading

#### **Exams**

There will be two in-class and closed-book exams for this class. The exams will be based primarily on the materials covered in class but will include research type questions as well.

### **Grading**

10%	→ hardco
25%	
30%	
35%	
	25% 30%

→ hardcopy required, no late submission, keep for 1 week



Letter grade will be assigned approximately as follows:

80% — 100%	A, A-	(truly exceptional, exceed expectations)
70% — 80%	B+, B, B-	(meet expectations)
60% — 70%	$C_{\perp}$ $C$ $C_{-}$	

Below 60% F



<sup>\*\*</sup> Final distribution may be adjusted based on the class performance.

<sup>\*\*</sup> Students who do NOT take exam(s) or miss project presentation will receive an "F" grade.

<sup>\*\*</sup> Individual exceptions to the scheduled exams should be both "individual and rare".

### Resources



#### **Computing Resources**

OSS 327 Computer Lab,

You can use any tools of your choice, but test it here before your project presentation.

Please check your UST e-mail account regularly.

#### **Support Staff**

Instructor Chih Lai for questions regarding the materials covered in class, design and implementation clarification.

GPS Lab assistant Marius Tegomeh (962-5517, mntegomoh@stthomas.edu) for questions on using the equipment in Room 327.

#### **Attendance Policy**

Course attendance is expected, but no grade is given for it. Students who miss sessions are responsible for all information in that session. Students who need to miss **presentations** or **exams** due to unavoidable conflicts must arrange in advance to make up the session with the instructor.

#### **Course Assignments**

Homework will be assigned from time to time during the semester in order to reinforce the concepts/techniques discussed in the class. Assignments will be collected on the specified due dates.

NO late submission will be accepted without proper reasons.

# **Mandatory College Policy Statement**

#### ENHANCEMENT PROGRAM FOR DISABILITY

• Classroom accommodations will be provided for qualified students with documented disabilities. Students are invited to contact the Disability Resources office about accommodations for this course. Telephone appointments are available to students as needed. Appointments can be made by calling 651-962-6315 or 800-328-6819, extension 6315. You may also make an appointment in person in Murray Herrick 110. For further information, you can locate the Disability Resources office on the web at <a href="http://www.stthomas.edu/enhancementprog/">http://www.stthomas.edu/enhancementprog/</a>.

#### ACADEMIC INTEGRITY

• Students are obliged to refrain from acts that they know or, under the circumstances, have reason to believe, will impair the integrity of the university. Violations of academic integrity include, but are not limited to, cheating, plagiarism, unauthorized multiple submissions, knowingly furnishing false or incomplete information to any agent of the university for inclusion in academic records, and falsification, forgery, alteration, destruction, or misuse of official university documents or seal. Any of these violations may cause the expulsion of the student from the program.

#### DISCLAIMER

• This syllabus is not a contract, but a plan for action. The instructors reserve the right to alter its stipulations, upon prior notification to students, if and when educational circumstances warrant changes.

### Chih Lai, Ph.D.

# Short Biography

#### Interests

Data mining on general & multimedia data, Healthcare/Biomedical Informatics

#### ■ Teaching Experience:

Dr. Lai is an associate professor with GPS. He has taught courses in Data Mining, Multimedia Information Retrieval, Healthcare Informatics, Real-Time Systems, and Software Engineering. He was also a visiting professor of the Informatics Department at Trier University of Applied Science in Germany in 2010. Dr. Lai also taught an Operating System course at the Computer Science Department of Oregon State University.

#### Research and Publications:

Dr. Lai's research interests include Data Mining, Multimedia Information Retrieval, Healthcare Informatics. Dr. Lai has published many technical papers on international conferences / journals sponsored by IEEE and ACM. Dr. Lai is the 2004/2011 University Research Grant recipient. Please visit <a href="http://personal1.stthomas.edu/clai">http://personal1.stthomas.edu/clai</a> for details.

#### Industry Experience:

Before joining University of St. Thomas, Dr. Lai was a principal software engineer, working on a next generation aircraft collision avoidance system (ADS-B) approved by FAA. Dr. Lai received three U.S. patents and three European patents, all related to aircraft collision avoidance algorithms. Dr. Lai also works with Medtronic and has pending patents on monitoring and evaluating Parkinson patients. Other industry experience includes building a network gateway between IBM / Novell networks.