CS3215: Software Engineering Project

CS3215, LN set #1: Course introduction

Course instructors:

Xue Yinxing, PhD student and TA

- In charge of course co-ordination and tool infrastructure

Markus Kirchberg, Adjunct Prof, Research Fellow at I2R

Ashwin Nanjappa, PhD student and TA

Stan Jarzabek, course instructor

Wish you most successful project experience!

Why software project course?

Challenges of software development in industry:

- Functional and quality requirements:
  - Production quality is not "just to get a program run"
  - Performance, maintainability, reusability
- Quality documentation (internal and external)
- Bigger project size:
  - Architecture, design principles
  - Work in teams
  - Development process (SDLC)
  - Project planning
- *Pressures*: deadlines, develop fast!

What's important in CS3215?

- Design is the main challenge in software development
  - Architecture: a bridge between user requirements and code
- Architecting software based on SE principles
  - How to apply SE principles in practice?
  - Analyzing the design, making good design decisions
- Software qualities: reliability, ease of change
- Incremental development process
- Team project: Learn to work with others

➢ Consolidate what you learned in earlier courses

➢ We expect our grads to do well in industrial projects!

How to impress you employer?

- Analytical, design and development skills
  - Formulate and analyze a problem at the concept level
  - Work towards the solution: design & implement
- Working knowledge of SE principles and "best practices"
- Think and work independently, but as a part of a team:
  - Plan team project, split work, plan your tasks
  - Effectively communicate among team members

What do you do in CS3215?

- Team project – form teams of six students
- Study problem description – requirements
  - Analyze on paper, play with models
  - Two weeks, Assignment 1
- Design and document software architecture
  - Subsystems, main components, interfaces
  - Reference point for design decisions and implementation
  - Build a prototype (learn basics of C++)
  - Two weeks, Assignment 2
- Complete project in three development iterations
  - Agile development processs

Course overview

- Problem description in the Handbook
- Lectures
  - explain the problem, principles and methods
  - guest lectures bring industry perspective
- **Consultations:**
  - **each team has one-hour slot per week**
- Five assignments guide you through the project
- Final project report
- Project presentation

Grading policy

- 20% assignments #1, #2, #3, and #4
  - Each assignment 5 marks - if you submit a reasonable assignment on due date
  - 4 marks - if your work or attitude is unreasonable
  - 0 marks, if:
    - You submit assignment late without valid reason
    - You do not work on the project
    - You got 4 marks in the last assignment and you did not improve
- **80% final project report, presentation, demo**

You are not panelized on the way as long as you work hard and deliver quality result at the end

Contribution to a project

- All team members must get involved and contribute to the project
- Any problems – please let us know, so that we help you resolve them
- Peer review: Each team member will be asked to provide feedback on other team members:
  - at the end of the first iteration
  - at the end of the course

Project evaluation and grading

- "having a program run" is not enough
- we expect at least:
  - implementation of functionality as described in Handbook
  - reliability: programs must pass our test benchmark
  - high quality of documentation and report
  - flexibility and reusability
- to get above B+ :
  - demonstrate some degree of innovation in terms of design solutions, query evaluation strategy, etc.
  - demonstrate maturity of skills in areas of team work, design, incremental development, approach to testing

Policy on project work

- you are permitted to discuss project with anyone
- solutions you hand in should be your own work
  - coding and documentation should be the work of your team only
- you may not view any code and document written for CS3215 by anyone not in your team, including past students
- you may not reveal your code and document to any students not in your team
- any case of academic misconduct will be prosecuted to the fullest extent provided for by University regulations

Implementation language: C++

I know Java – why to learn yet another language?

- Each language is different, you may use may programming languages in your career
- Many large software systems are implemented in languages other than Java
- CS graduate is expected to learn a new language fast

Learn C++ and increase your market value

C++ Development Environment

- Visual Studio 2010
  - IDE
  - standard C++ compiler with library (STL)
  - debugger
  - automated build tool
- other tools: SVN, cppUnit
- find out details via Tools link at course Web site

C++ books

C++ books: (in the Forum Bookstore)

- C++ How to Program, by (Harvey & Paul) Deitel & Associates, Pearson
- Data Structures & Other Objects Using C++, Michael Main, Walter Savitch

Additional references:

- any other C++ books, many can be found in the Library
- much info can be found on the Web
- check details on the course Web site

First two weeks in glace:

- Register teams (check the course Web site)
- Download project materials from the Web:
  - Project Handbook, Lecture Notes, Assignment 1&2
  - Report Format for Assignments and Final Project Report
- Study Project Handbook and do analysis exercises described in Assignment 1
- Read paper describing CS3215 philosophy and experiences (link on the course web site)

S. Jarzabek and P.K Eng " Teaching an Advanced Design, Team-oriented Software Project Course ," Proc. 18th Conf. on Software Engineering Education and Training (CSEE&T), April 2005, Ottawa

- Start learning C++

--- The End ---