COP701: Software Systems Laboratory

Introduction

Why this course?

Gain experience writing large software systems

- Practice designing architecture of a software system
- Tools and practices for software development in teams
  - Version control
  - Build automation
  - Unit testing
  - Autogenerated documentation

Course format

This is a practical-oriented course (credit structure 0-0-6)

- No weekly lectures, only once in a while
- 3 large assignments, one each month
- Other things you should learn on your own while doing the assignments
- **Sign up for Piazza** (link on course webpage), access code “cop701”

Recommended books: *Head First Design Patterns* and *Beautiful Architecture*

Assignments

Design and implement a specified medium-to-large sized software application

- Form groups of 2-3 by end of Monday. For those left over, we will make groups
- Any programming language is OK
- Designing, coding, testing will take time. Use the whole month given to you!
- Submit both (i) code, and (ii) report about your system design & work division

Assignments (cont.)

Recommended strategy:

- Sit together with group and design the software architecture
- Then go implement modules individually
- Write the report early (first week!) — so your own plan is documented. Update it at the end if it changed

Tools and practices

Version control, build automation, unit testing, autogenerated documentation

- You have to learn these yourself (lots of great online resources available!)
- Learn and apply any 2 in Assignment 1, any 3 in A2, all 4 in A3

Make sure you have gone through the whole course webpage by Monday!

One last requirement: not being slow at typing :)

Plagiarism policy

Don’t do it!

The end.

Plagiarism policy

Plagiarism = presenting someone else's work as if it is your own.

- **Don't copy** from others' code, from others' reports, from sites online
- **Don't even "copy then change".** If you need to learn from someone else's work, read and understand it, then write it yourself without looking at the original again
- **Always credit your sources:** third-party libraries, ideas from discussions or online sites, etc.

Assignment 1: Your own Wikipedia

Wiki = collection of easy-to-edit pages with links between them

Implement a GUI application that provides a wiki.

- User can view pages, click links, create/edit/delete pages
- Pages should be stored in plaintext as Markdown files
- Should run as a desktop application (not on web server like Wikipedia)
- Detailed requirements on course webpage

Markdown

```
Markdown
========

**Markdown** is a simple way to
write *formatted text*.

Its [syntax](markdown_syntax) is
easy to learn.
```

Markdown

Markdown is a simple way to write *formatted text*.

Its syntax is easy to learn.

Assignment 1

So, you will need to do:

- GUI programming
- Markdown parsing
- Text editing
- Formatted text display
- Filesystem handling
- etc.

…and any 2 of the following:

- Version control
- Build automation
- Unit testing
- Autogenerated documentation

This is why you are in groups :)
Good luck!