The primary objective is to provide additional (advanced/smart) software engineering skills that may be of use to you in your future projects.

The secondary objective is to reinforce your mastery of software engineering fundamentals.

The teaching approach is primarily PBL.

You will be given a project that asks you to demonstrate understanding of the advanced concepts, and ability to use them in solving a real-world problem.

You will have an exam that asks you to demonstrate a critical analysis of the interaction between these concepts.
CSC7336: Overview

http://www-public.int-evry.fr/~gibson/Teaching/CSC7336/

Advanced Software Engineering for Smart Devices

The material will be uploaded dynamically; the teaching approach is based on PBL and much of the learning will be through interaction/group work during the assigned lecturing time. (Please check the website for updates before every lecture.)

Assessment

The assessment will involve:

- project work - deadline Friday 9th February (month)
- written exam (DATE TO BE DECIDED)
- Example Exam Questions

Each will be marked out of 20. The weighting between each will be decided based upon performance in each. 75% of the final mark will be based on the best mark from either the project or the exam. The remaining 25% will be based on the mark in the other assessment.

Useful Links

- html Java Monitor integrated with Eclipse

Sessions

Sessions are a mix of problem-based learning, group project work, directed practicals, interactive lectures and traditional lectures. There is no preset format - the lecturer organises the style and content of each session depending on the needs of the class.

Session 1: Wednesday 8th November (9h45, B08) - Introducing Aspects

Lecture Slides

Introduction To CSC7336 slides, questionnaire.pdf

A problem with aspects.pdf, SegmentOverlap Java project to import into Eclipse

Additional reading material

- pdf Aspect-oriented programming, Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Lopes, Jean-Marc Leitinger and John Irwin, published at ECOOP97.
- pdf The paradoxical success of aspect-oriented programming, Friedrich Steimann, published at OOPSLA06.
Your Learning Objectives?

You have to meet my learning objectives

But I can also match these to your learning objectives

However, we do need to assess what you have learned
Prerequisites:

• Foundations of software engineering,

• Foundational mathematics,

• Object oriented Programming
**Things We Will/May**

**Aspects** - AOP and AOD - *how do we distribute development increments?*

**Reflection** - *how do we do self-aware?*

Foundations to **AI** - *how do we do smart?*

**Distributed Algorithms** - *how to cope with the major challenges?*

**Big Data** - Analysis techniques and Training Expert Systems

Parallel (multi-threaded) Programming -

Cloud Services - Google, Amazon, IBM

**App/Game Design** - rapid prototyping

**Simulation modelling**

Developing for Android / Developing for iOS

**Different programming languages** (like Wyvern, Prolog, Clojure, Ruby, ...)

.... Do you have any ideas/requests???
Please complete the questionnaire

GET ALL THE INFORMATION YOU CAN, WE'LL THINK OF A USE FOR IT LATER.