This is an introduction and overview for the lesson and assignments part of the SE course.

More information available from the course page at:
http://www.idt.mdh.se/kurser/cd5360/02/labs
Personal

- **Lessons**
  - Andreas Sjögren, [ase@mdh.se](mailto:ase@mdh.se), room 373

- **Assignment supervisors**
  - Nerina Bermudo, [nbo@mdh.se](mailto:nbo@mdh.se), room 387
  - Rikard Land, [rld@mdh.se](mailto:rld@mdh.se), room 373
  - Andreas Sjögren

Assistants for the assignments.
Lessons and Assignments

- **Purpose**
  - Repetition and application of important concepts from the lectures
  - Basic UML knowledge
  - Preparation for the project

- **Organization**
  - Lessons once a week
  - Assignments in groups of 1–2 students
  - Computer room booked 8h week (v408–lab6)

Contents of the lessons:
Modeling:
Requirements Engineering (with UML)
Design (with UML)
In addition:
Configuration Management (and tools)
Project Planning (for the project)
Software Engineering

- An engineering discipline concerned with all aspects of designing and developing high quality software, from the early stages of specification, through maintenance of the system after it gone into use, and finally the retirement of the system.

- The primary goal of a software development is to produce high quality software that satisfies evolving needs, everything else is secondary.

- But, “secondary” does not imply irrelevant!

“Secondary”, but very important:
Exposé the requirements of a system
Define the architecture of the system
Design the system.
Development process.
Tools, people, right knowledge, training, etc.

We are concerned most of the early stages of specification and modeling in this part of the course.
The models and the specifications are also an important source in the maintenance phase, since there are a source of documentation of the system, but only if they are kept up-to-date with the software.

The importance of modeling increases when the complexity of the intended system increases.
The Unified Modeling Language (UML)

- Unified
  - Combination of the modeling languages Booch, OMT, and OOSE

- Modeling
  - Visualization
  - Specification
  - Construction
  - Documentation

- Language
  - Common Vocabulary
  - Syntax and Semantics


U:
Exist (or at least in the beginning of the 90’s) several different modeling languages. Three well-known unified into UML.

M:
Originally designed as a language for OOA/OOD with an assumption of an OO-implementation of software. Now, it is object oriented, but it can be used for a more general purpose.
If we for instance modeling a software system, the model is still object oriented, but the final implementation does not need to be object oriented.
We can modeling other things than software

L:
Vocabulary – has set of textual and graphical symbols
Syntax – rules for how to combine the symbols into diagrams
Semantics – the meaning of syntactical correct diagrams, non-formal and based on a meta-model

UML is designed to be extended, the (major) syntax element for this is stereotypes.
The most important artifact in UML is the models. Sometimes they nevertheless are overlooked. Often people think treats the diagrams as the most important. The 9 models intend to cover all the important decisions that go into building and documenting a system.

1 – the organization (sometimes the word business are used for the domain as well.)
2 – the context of the system
3 – the system’s functional requirements
4 – an idea design (optional)
5 – the vocabulary of the problem and its solution
6 – the system’s concurrency
7 – hardware topology
8 – parts used to assemble and release the system
9 – paths by which the system is validated and verified

We will not cover all this models in the lessons. We will only discuss the models which are used in the Requirements and Design phases. This models are mainly model 2-5.
UML is not a Method

- **Modeling Language**
  - Mainly a (graphical or textual) notation that a method uses to express requirements and design

- **Development Process**
  - Describes the major activities, as series of steps to take

- **Software Development Method**
  - Consists of a modeling language and a process

Methods like, for instance, RUP (Rational Unified Process). This definition is not “standard”. Most often, the word method and process is used for the same thing. (for instance is the method RUP named ‘Process’).
Building Blocks in UML

- **Things**
  - Main building block
  - Denoting physical or abstract elements, classes, interfaces, etc
- **Relationships**
  - Connections between things
- **Diagrams**
  - Groups of interesting collections of things which captures different aspects of a model
  - Easy to overestimate the importance of the diagrams

The basics of UML are defined in a conceptual model, with three different elements.

**Things:** Main building block

**Relations:** Connections between things, like generalization, and dependencies of different kind.

**Diagrams:** Organizational information

Things and relations is enough for modeling. The diagrams are used to clarify and capture different aspects of the models.

We could spend a lot of time on this conceptual model, since it will be reflected in all UML models. But we will not discuss this as much further. Instead we will be more pragmatic and look into how too use UML, and hopeful you will be able to see the common concepts from this.
We use Rational Rose in the course

Remember that Rose is a modeling tool, not a drawing tool, some features will seem rather un-logical if you treat it as a drawing tool.

For instance, if you delete something from a diagram it is not sure it will be deleted from the model. One thing can be included in many diagrams.

Rational Rose is a quite expensive application, but it is available in a free limited student version.
Lessons, more detailed contents

- Use cases
- Use case diagrams
- Identify domain concepts – class diagram (conceptual)
- Activity diagram
- Design concepts and relations – class diagram (specification)
- Architecture – class diagram
- Temporal relations between classes – sequence diagram
- Internal states in classes – state charts

The contents of the lessons and assignments, with UML concepts. Just a brief look of what we will talk about in all the lessons. Abstract to talk about without relying on what all the concepts are.
Assignments

- Assignments and more instructions available from the course page:
  http://www.idt.mdh.se/kurser/cd5360/02/labs

- Reporting the assignments:
  - Mail your lab assistant (see course page)
  - Name your files and mails properly
  - Every assignment should have an additional textual description
  - Deadline at 12.00 the day for next coming lesson
    (for lab1: 29 Jan 12.00, etc.)

Send the file to your lab assistant (following the naming conventions described on the main lab page), together with a written comment.

The purpose of the comment is to show that you actually know what you have done.

It should describe and motivate any major decisions you have taken.

We don't want a complete translation of diagrams into text, but rather a discussion of which parts that could have been modeled in a different way, and why you decided not to.

If there is many errors or problems with your solution we want you to discuss it in the computer room and show it on the computer instead.

It is really hard to discuss these assignments through email (both for you and for the assistants). So do not get frustrated if you just receive a “not approved” as answer on your solution, that means you should come to the next scheduled lab and we will discuss it there. (or contact your assistant at the office some other time if you cannot come to the lab).
Literature

- No book for this part of the course
- The presentations will be available from the course page
- UML books
  - UML Distilled (2nd ed), M. Fowler, Addison-Wesley
  - The Unified Modeling Language User Guide, Booch, Rumbaugh and Jacobson, Addison-Wesley

We do not use any particular book for this part of the course. There are a lots of different books about UML. The user guide is from original designers and is a rely extensive guide to UML. The distiller book is a rely good first introduction to UML. Rather short and very well written.