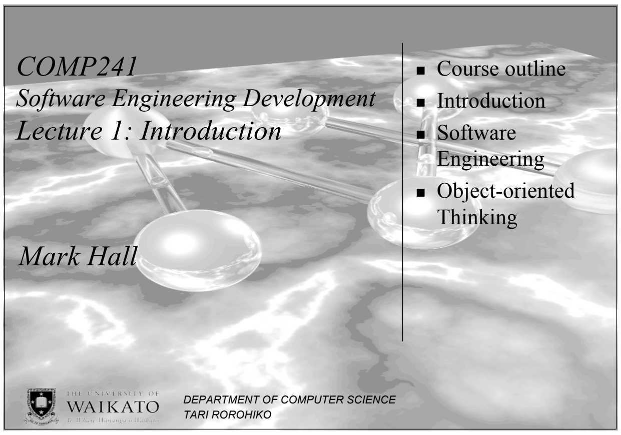


COMP241

Software Engineering Development

Lecture 1: Introduction

Mark Hall

- Course outline
- Introduction
- Software Engineering
- Object-oriented Thinking

THE UNIVERSITY OF WAIKATO

DEPARTMENT OF COMPUTER SCIENCE
TARI ROROHIKO

Lecturer

- **Mark Hall**
  Room G 1.05
  `mhall@cs.waikato.ac.nz`

Assessment

- **100% internally assessed**

<u>3 Assignments</u>:

1.	Testing	**20%**	due Friday 23 March 2007
2.	OO Programming	**20%**	due Friday 27 April 2007
3.	GUI Programming	**20%**	due Friday 1 June 2007

<u>2 Tests</u>:

Test 1	**20%**	Thursday 5 April 2007
Test 2	**20%**	Thursday 31 May 2007

Computing Resources

- Java 1.5.0 and 1.6 have been installed on the Linux Systems in Labs 6 and 7 in R-block.
- You can download the software from SUN:
  `http://java.sun.com/`

Also Used:

- JUnit – `http://www.junit.org`
- ANT – `http://ant.apache.org`
- CVS – `http://www.cvshome.org`

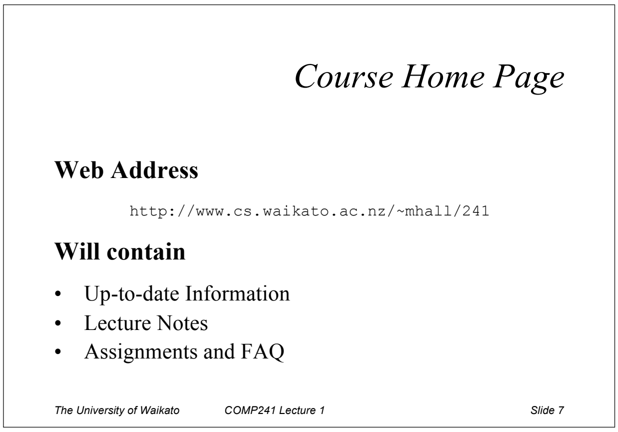
Textbook

- Cay Horstmann:
  Object-Oriented Design & Patterns.
  Wiley, 2006.

Object-Oriented Design & Patterns
SECOND EDITION
CAY HORSTMANN

Texts on Java

- Kathy Sierra, Bert Bates. *Head First Java.* O'Really, 2003.
- Timothy Budd. *Understanding object-oriented programming with Java*, Updated Edition. Addison-Wesley Longman, 1999.
- Roger Duke, Eric Salzmann. *Java Genesis*, Addison-Wesley SprintPrint, 2000.
- Bruce Eckel. *Thinking in Java*, Prentice-Hall, 2002.
  `http://www.mindview.net/Books/TIJ/`
- Java Home Page.
  `http://java.sun.com/`



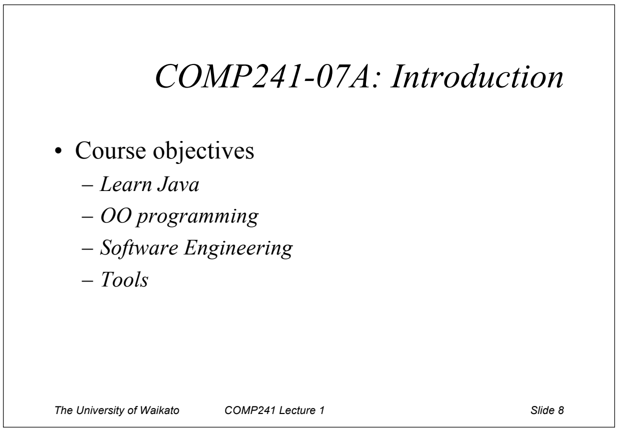
Course Home Page

Web Address

`http://www.cs.waikato.ac.nz/~mhall/241`

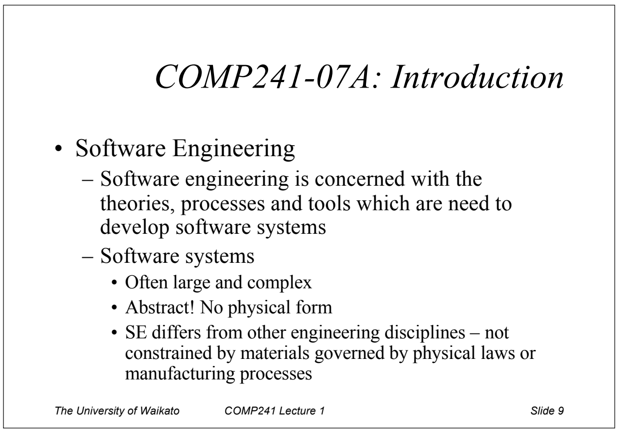
Will contain

- Up-to-date Information
- Lecture Notes
- Assignments and FAQ



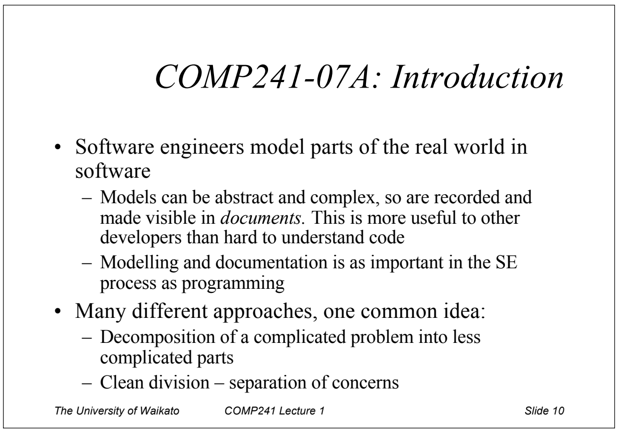
COMP241-07A: Introduction

- Course objectives
  - *Learn Java*
  - *OO programming*
  - *Software Engineering*
  - *Tools*



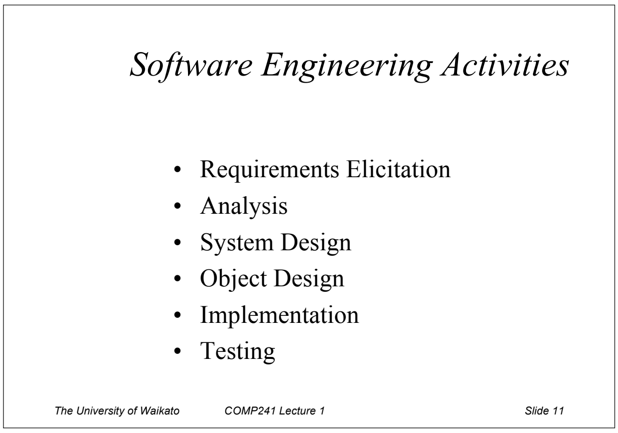
COMP241-07A: Introduction

- Software Engineering
  - Software engineering is concerned with the theories, processes and tools which are need to develop software systems
  - Software systems
    - Often large and complex
    - Abstract! No physical form
    - SE differs from other engineering disciplines – not constrained by materials governed by physical laws or manufacturing processes



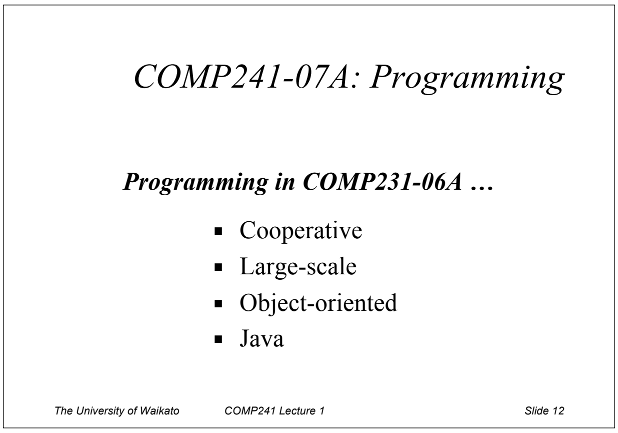
COMP241-07A: Introduction

- Software engineers model parts of the real world in software
  - Models can be abstract and complex, so are recorded and made visible in *documents.* This is more useful to other developers than hard to understand code
  - Modelling and documentation is as important in the SE process as programming
- Many different approaches, one common idea:
  - Decomposition of a complicated problem into less complicated parts
  - Clean division – separation of concerns



Software Engineering Activities

- Requirements Elicitation
- Analysis
- System Design
- Object Design
- Implementation
- Testing



COMP241-07A: Programming

Programming in COMP231-06A …

- Cooperative
- Large-scale
- Object-oriented
- Java

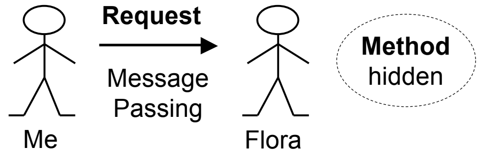
Object-Oriented Thinking

Objective

- Not just learn Java, but also *understand* why it is the way it is.
- Learn the object-oriented *worldview*.

A Way of Viewing the World

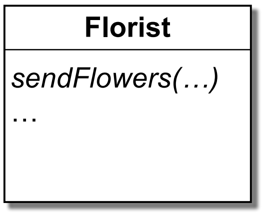
Sending flowers to another town



Delegation:

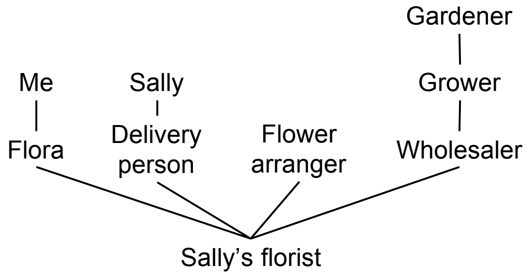
"Don't even want to know the details."

Agents and Services



"Start looking at the **service** an object can provide for us."

Agents and Communities



Computation is Simulation

Instead of a bit-grinding processor … plundering data structures, we have a universe of well-behaved objects that courteously ask each other to carry out their various desires.

[Ingalls 1981]

Responsibilities

Ask not what you can do to your data structures, but what your data structures can do for you.

"Act on a data structure"
vs.
"The data structures act on themselves"