

Software Tools

Introduction to Part II

Part II - Lecture 1

Christof Lutteroth

2009
YEAR

COMPSCI 732

The University of Auckland | New Zealand

- Became a permanent lecturer last year
- From Berlin, Germany
- First time for me to give 732 (looking forward to it!)
- My research interests: model-based SE, HCI, DBMS, computer graphics, ...
- Contact details:
 - lutteroth@cs.auckland.ac.nz
 - Phone 373-7599 84478
 - Office 303 - 494 (4th level CompSci building)
- If you have questions, come to my office at any time
- A good time to see me is directly after the lectures

Part II Timetable

| When | | What | Where |
|---------------------------------|-------|------------------------------|------------------------|
| Monday | 13-14 | Lecture | 279 |
| Tuesday | 13-14 | Lecture/Lab | 279 GPL (BYO chair) |
| Thursday | 13-14 | Paper Session | 279 |
| Once every week (you decide) | | Your project team meeting | You decide |
| 4 th June | 7pm | Assignment 2 (25%) | ADB |
| TBA | | Exam (50%) | TBA |

2009

YEAR

COMPSCI 732

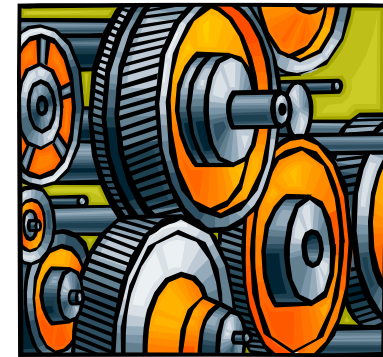
The University of Auckland | New Zealand

Introduction to Part II



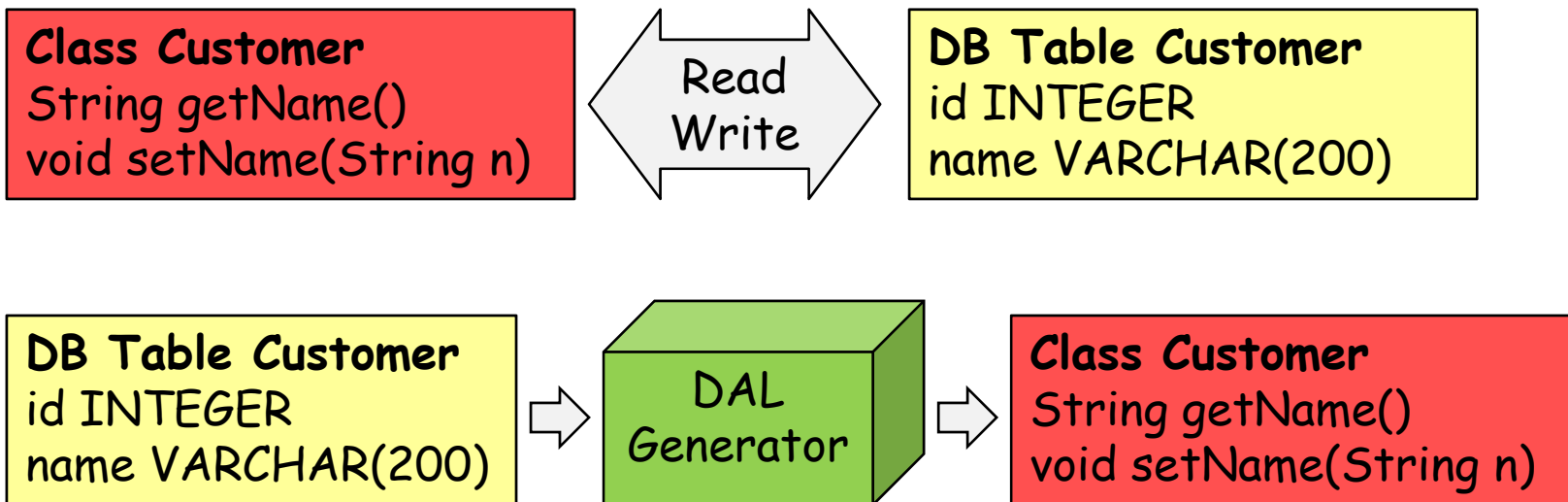
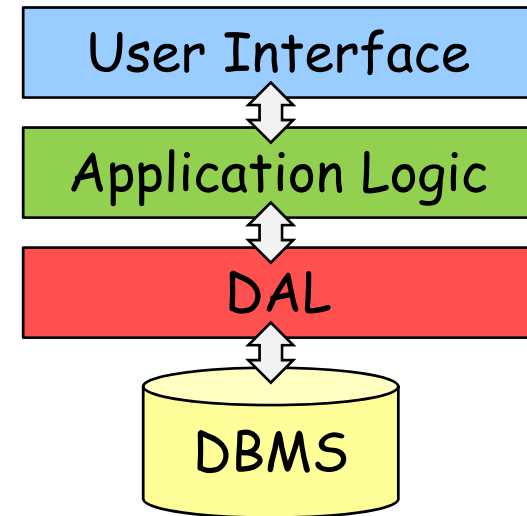
Software Tools

- Humans are necessary for creative, intelligent tasks
- Tools can **support** such tasks
 - Increase productivity with useful functionality
 - Guide the developer (e.g. context help)
 - Avoid defects
- Humans are not necessary for highly repetitive, routine work
- Tools can **automate** such tasks
 - Increase productivity; more time for creative work
 - Avoid defects introduced by the human factor



Data Access Layer (DAL)

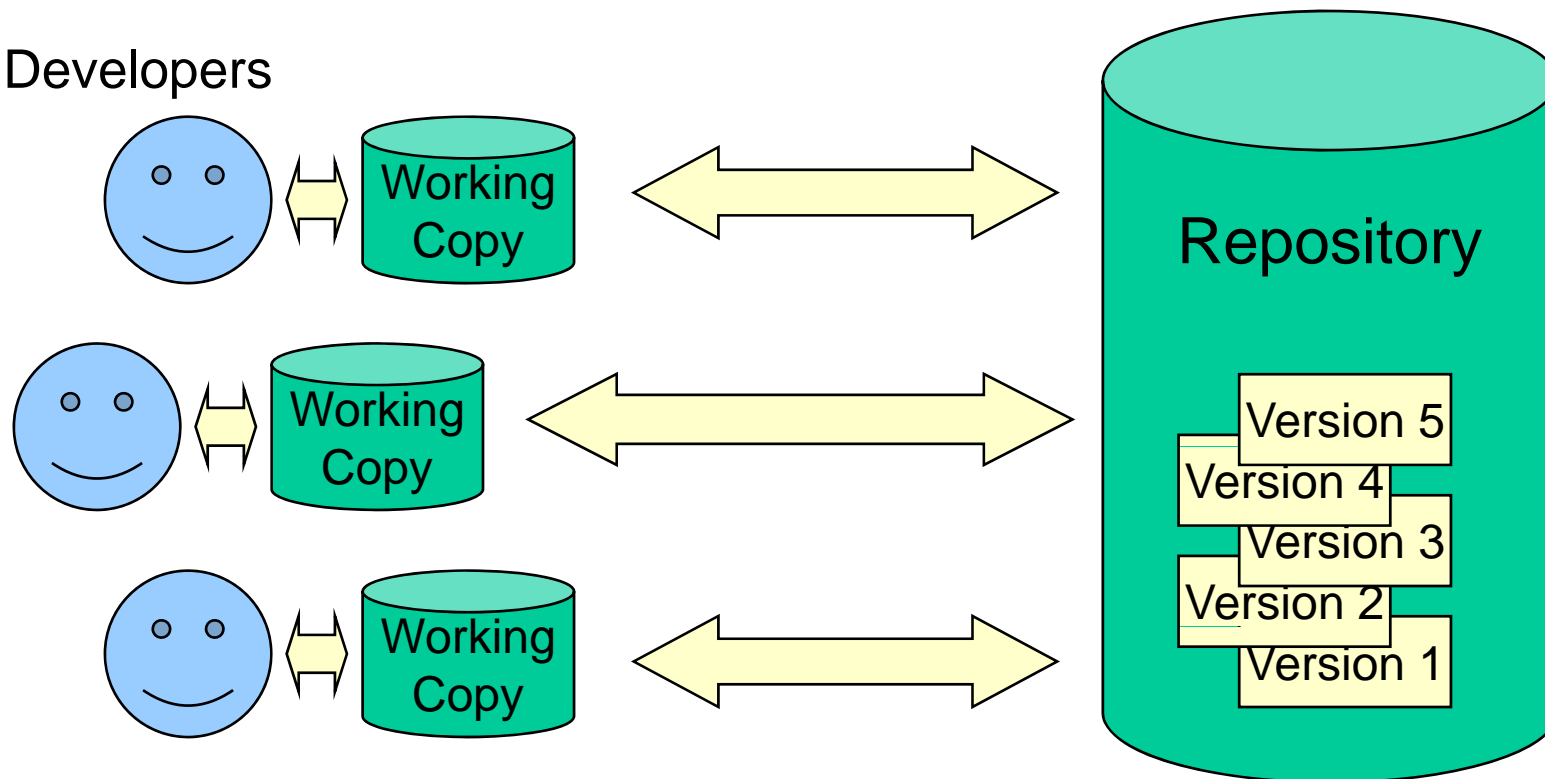
- Application layer that provides functionality for convenient DB access
- Enables the use of OO classes to read and write from/to the DB (instead of having to use SQL)



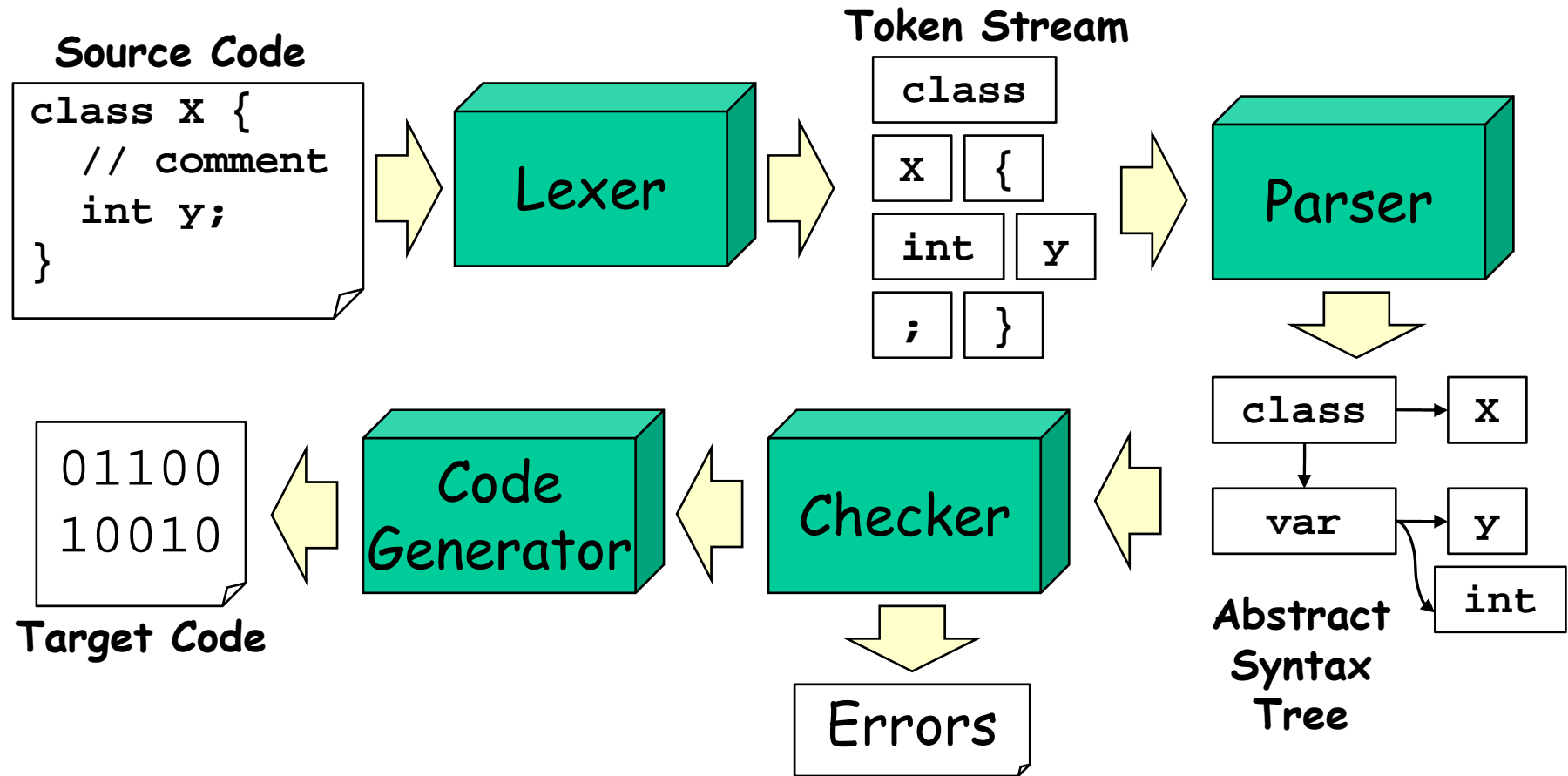
Version Control Systems

- Technology to manage changes that several developers do on a common repository
- Changes create new version of the changed files
- Old versions are always accessible

Developers



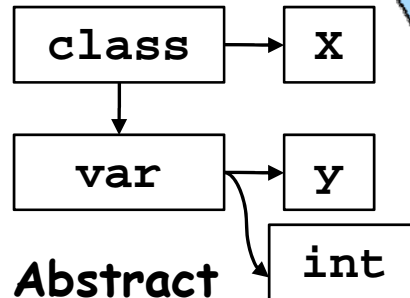
Compilers



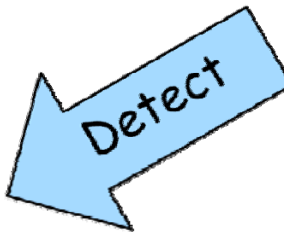
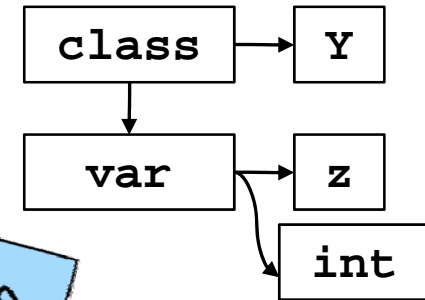
- Lexer chops the source code into tokens
- Parser constructs the syntactic relations between the tokens (abstract syntax tree, AST)

Processing Source Code

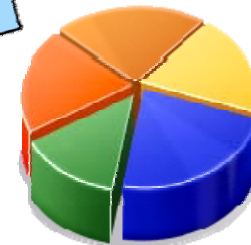
```
class X {  
  int y;  
}
```



Abstract
Syntax
Tree



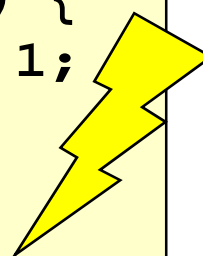
Errors
Warnings



Type Systems

Type Checking: detect potential runtime errors in source code

```
int m(String s) {  
    int y = s + 1;  
    m(y, 3);  
    return s;  
}
```



Type System: Formalize type checking by using rules that describe correct programs

(Expr Plus)

$$\frac{\Gamma \vdash E_1 : Nat \quad \Gamma \vdash E_2 : Nat}{\Gamma \vdash E_1 + E_2 : Nat}$$

(Expr NotEq)

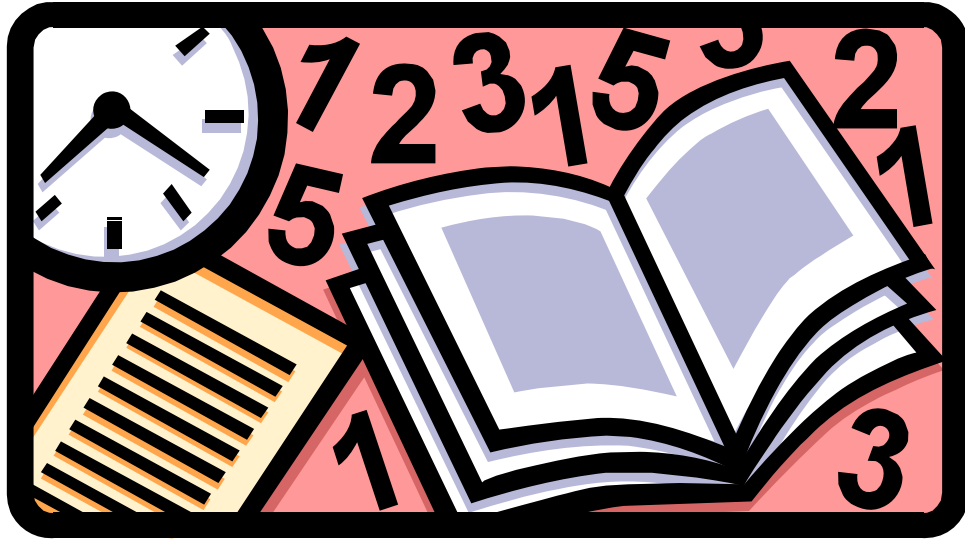
$$\frac{\Gamma \vdash E_1 : Nat \quad \Gamma \vdash E_2 : Nat}{\Gamma \vdash E_1 \text{ not} = E_2 : Bool}$$

Learning Outcomes

After the course you should be able to...

- Describe what data access layers (DAL) are, why they are important and how they can be developed
- Create your own models in the PDStore system and use them in Java
- Describe the main concepts of version control systems
- Use Subversion to efficiently work in a team
- Use ANTLR to create your own lexers and parsers
- Write programs that generate or analyze source code
- Do simple type derivations using formal type systems
- Present academic research papers

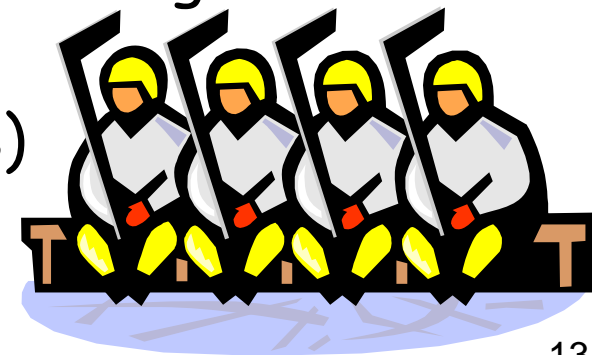
Assignment 2



Assignment 2

25% in total, split up over three graded parts:

- Research paper presentation in teams of two (5%)
 - Presentations every Thursday over the next weeks (starting 7th of May)
 - 4 mins each + 2 mins questions
 - More in the next lecture
- Projects in teams of four
 - Project implementation, together but graded individually (10%)
 - Project report, individually (10%)
4 pages including figures



The Project



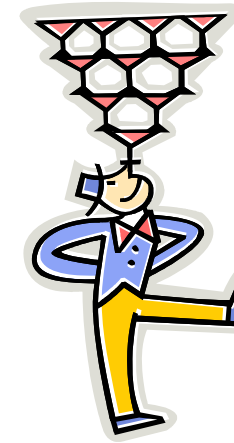
- Teams of 4
- One language grammar given per team: Java, C#, C++, JavaScript, Python, Ada, Pascal, HTML, CSS, OCL, ... (first come first served)
- Choose yourself from <http://www.antlr.org/grammar/list>
- Develop a
 - Data model: enables storage of source code in a database
 - Lexer and parser: analyze source code and store it
 - Printer: read code from the database and print it as text files
 - Do some cool analysis, metric or visualization

Tools to Use

- Java
- Eclipse IDE
- PDStore
- Firebird DB
- Subversion
- ANTLR



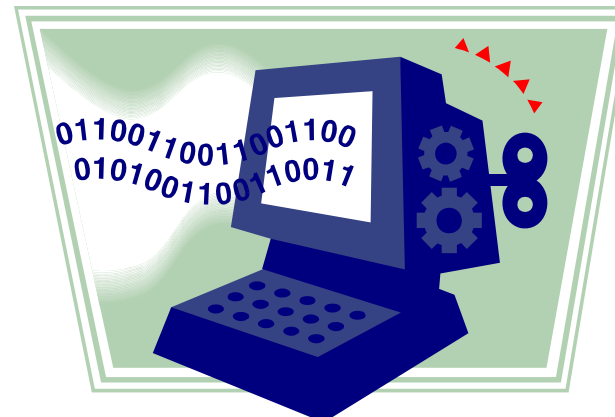
Project Expectations



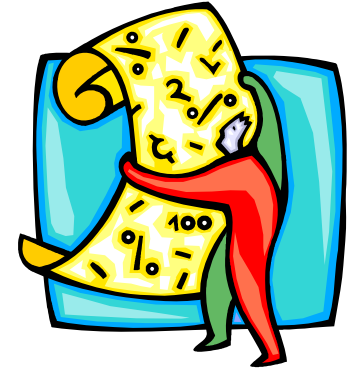
- The project size is scalable (you need not cover the whole language)
- Expectations:
 - Work together as much as you can (you can work with other teams as well!)
 - Have a project group meeting every week
 - Time spent per week per person on 732: 10 hours
 - Come to the lectures/labs (you will learn what you need to do a good project)
 - If you are stuck, ask! (your teammates, other teams, the lecturer)
 - Only the project report has to be written individually

Project Grading Schedule: Implementation (10%)

- Were the given tools used (ANTLR, PDStore, SVN)?
- Has everybody contributed adequately?
- Was an appropriate data model defined?
- Was an appropriate parser generated that stores source code in the database?
- Is there a printer that can read source code from the database and print it into text files?
- Are there some nice things?
e.g. search, analysis, metrics



Project Grading Schedule: Report (10%)



Approx. 4 pages (including figures)

- Have you introduced the project and its aims?
- Have you briefly introduced the tools that were used?
- Have you described the current state of the project?
- Have you described the work you contributed to the project?
- Have you described the challenges of the project?
 - What was really easy to solve
 - What was hard, i.e. took the most time
- Have you described future work, i.e. what needs to be done (unfinished work)? What could be done?



Summary

Today's summary

- Part 2 will cover various types of tools and techniques:
 1. Data Access Layers (DALs)
 2. Version Control Systems (VCSs)
 3. Compilers
 4. Source code processing techniques
 5. Type systems
- In Assignment 2 you will do
 - A research paper presentation in teams of 2
 - A source code processing project in teams of 4

**Form a team, pick a language to process,
and send me an email with your team
details and language choice**

Quiz

1. How can software tools help with repetitive routine tasks?
2. How can software tools help with creative tasks?
3. Name four of the five main topics covered in part 2.