Final Project
Advanced Programming
Planning + Ideas

Paul Klint
# Planning

<table>
<thead>
<tr>
<th>Week</th>
<th>Tuesday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>9, Oct 30 &amp; Nov 2</td>
<td>Grammars and Parsing</td>
<td>Assignments + Grammars and Parsing</td>
</tr>
<tr>
<td>10, Nov 6 &amp; 9</td>
<td>Grammars and Parsing; Project Ideas</td>
<td>Assignments + Language Processors</td>
</tr>
<tr>
<td>11, Nov 13 &amp; 16</td>
<td>Language Processors</td>
<td>Assignments + Language Processors</td>
</tr>
<tr>
<td>12, Nov 20 &amp; 23</td>
<td>Present Plan 1-4</td>
<td>Present Plan 4-8</td>
</tr>
<tr>
<td>13, Nov 27 &amp; 30</td>
<td>Test II</td>
<td>Work on Project</td>
</tr>
<tr>
<td>14, Dec 4 &amp; 7</td>
<td>Work on Project</td>
<td>Work on Project</td>
</tr>
<tr>
<td>15, Dec 11 &amp; 14</td>
<td>Work on Project</td>
<td>A Software Engineering Perspective</td>
</tr>
<tr>
<td>16, Dec 18 &amp; 21</td>
<td>Present Results 1-4</td>
<td>Present Results 5-8</td>
</tr>
</tbody>
</table>
The Final Project

• Starts with selecting a topic and writing a brief project plan *which has to be approved.*
• Do the project.
• Report results.
• Is done in teams of 2-3 (this should be agreed upon before hand)
Project Ideas

A project should satisfy the following requirements

• It should be interesting
• It has a clear goal
• It has clear results.
• It can be carried out (mostly) with what you have learned in Advanced Programming
• It is not too ambitious
• It can be carried out in a 3 week period.
Idea: AUC Course Catalog

- Goal: automatic analysis and querying of the AUC course catalog.

- Topics (for possibly several projects):
  - Create a data model for the courses
  - Extract data from the available information and represent this in the data model
  - Define analyses on the data model
  - Visualize the data
  - Define queries on the data model
  - Create a user-interface to interact with the data
Idea: Analyze Open Data

- Goal: Analyze an open data source of your choice (examples: CBS, open government data, ...)

- Topics:
  - Convert the data in a format that can be read into Rascal (CSV, XML, HTML, ...)
  - Identify the questions to be formulated.
  - Describe the analysis
  - Report results:
    - HTML
    - Visualization
Idea: Programming Challenges

• Explore the problems and programming tasks listed at
  • http://rosettacode.org/wiki/Category:Programming_Tasks
  • http://programmingpraxis.com/contents/themes/
  • http://projecteuler.net/

• Select one that is sufficiently challenging (but also doable) and implement it in Rascal.
Expected Deliverables

• Short project plan (at start)
• Description of results
• Source code
• Presentation
• Demonstration
Short Project Plan

- Small (1-2 page) project plan consisting of
  - Name: a descriptive name
  - Synopsis: a one sentence summary
  - Objective: 1-2 paras on what you want to achieve
  - Approach: 1-2 paras how you want to achieve this
  - Expertise: what do you need to know to do this?
  - Risk: what could go wrong?
  - Results: what are the result you expect?

- This plan will be presented in class and should be approved by teacher!
Description of Results

- A short paper (max 5 pages):
  - Title
  - Abstract
  - Goal
  - Approach
  - Results
  - Limitations
  - What you have learned from the project
  - Used Literature or websites.
Source Code

- Write code that is as readable as possible
- Use consistent naming and layout
- Add comments but do not exaggerate
- Use a true Rascal-style by avoiding too imperative code
Presentation

• A 10 minute presentation about your project
  • Focus on goal and achievements
  • Not too many details.
  • Highlight what was interesting or unexpected
  • What did you learn?
Demonstration

- Give a short demonstration of the software that you developed in your project.
Grading

- The grading of your work will take into account
  - Complexity of the project
  - The results
  - Project plan
  - Paper
  - Presentation
  - Demonstration