

Advanced Programming

Course for CSE
winter term 2010/2011

Tobias Neckel

Why such a course?

- **Experience 1:**

- Class of CSE students is very heterogeneous
 - different university systems and traditions
 - different fields of previous studies
 - even in the case of identical study fields different strengths and specializations
- Programming skills range from “very experienced” to “have already seen a computer”

- **Experience 2:**

- Necessary to provide the “clean” fundamentals of Object Oriented Programming
- If exercises are tailored, everyone can benefit

Experiencing to Program

- **What kind of language?**

- scripting languages,
 - either stand-alone (perl, python, ...)
 - or embedded (Matlab, Maple):
straightforward, but limited in performance and, thus, relevance (especially for HPC applications)
- compiler languages (Fortran Pascal, C, C++, Java, ...):
powerful, but more complicated (syntax-dominated: `&*{}(())...`)

- **In CSE:**

- **AdvProg**: learn a “real” language (Java) during the first semester, and do it “objects first” – i.e. not the quick & dirty way!
- **Intro2SciComp**: program from the very beginning, but in a simpler and, basically, prototyping environment (matlab)

- **How can one learn to program?**

- by doing: that’s the way done in Intro2SciComp
- a bit more systematically: here in AdvProg

Why Java?

- **Language design:**
 - Clean implementation of general concepts
 - Rather straightforward (compared to C or C++)
- **Object-orientation:**
 - Widespread *programming paradigm* (which others do you know?)
 - Allows to focus on design ideas
 - Facilitates a software engineering perspective (what's that?)
- **Popularity:**
 - Widely used
 - A lot of literature (probably too much ...), tools, ...
- **Development environment available: Eclipse, BlueJ**
 - Simple user interface
 - Important features (visualization of UML-like class structure)
 - “Real objects first”: create objects and call methods as first activity
 - Not a constructive access to programming (“DO loop first”)

Tutorials – Concept

- **Responsible persons:**

- First contact: Your tutor in your exercise group!
- Second contact: Martin Roderus
 - Room: 02.05.058
 - Email: roderus@in.tum.de

- **Tutorials in small groups:**

- Tutors from last CSE class (last year's beginners in CSE)
- “**Supervised teaching**” – part of additional education we offer within our honours program “**Bavarian Graduate School of Computational Engineering (BGCE)**”
- Important! Learning to program in a lecture is a bit like learning to swim in a lecture 😊.

Tutorials – Organisation

- **Structure:**
 - Two tutorial blocks
 - Monday, 8:30 – 10:00
 - Monday, 12:15 – 13:45
 - Start: Nov. 8 (Nov. 1: “Allerheiligen” holiday)
- **Registration:**
 - For lecture/exam:
 - via TUMonline
 - If not done: Do it until Friday!
 - For tutorials:
 - Registration for lecture necessary!
 - Will be done by Martin Roderus
 - You will receive an announcement email