Programming III
2501ICT Nathan

René Hexel

School of Information and Communication Technology
Griffith University

Semester 1, 2012
Outline

1. Course Organisation
2. Course Content
Teaching Team

- **Lecturer**
  - René Hexel (*r.hexel@griffith.edu.au*)
  - Use course code (**2501ICT** or **7420ICT**) Subject for eMails!
  - Technology Building (N44) Room 2.21

- **Tutors**
  - Carl Lusty
  - Hans Wannop
  - Available in Labs and Tutorials
Teaching

- Lectures (2 hours each)
  - Mondays 8–10am, N44_0.21
  - Fridays 2–4pm, N13_0.05
- Labs (2 hours)
  - start in week 1!
  - N44_2.25 and 2.34
  - as enrolled (Tuesdays 2–4pm or Wednesdays 4–6pm)
  - assignment milestones and feedback
- Drop-In Tutorial (1+ hours)
  - Fridays 2–4pm, N44_2.25 and 2.34
  - help and exercises
  - announced in the Lecture and on the Web Page.
Labs

- **Tutor Assistance**
  - Ask Questions!
  - Programming Practice

- **Part of the Assignments**
  - Necessary skills to complete Assignments
  - Programming Environment (Compiler, Makefiles, Subversion, ...)
  - **Milestones are due in the lab each week!**
  - **Come prepared!**

- **Outside official hours**
  - Labs close at 11pm
  - Dwarf is accessible via VLink from home!
  - **Most people will need to spend appx. 10 hours / week on the course!**
Assessment

- 2 non-trivial Assignments
  - Assignment 1 (20%), due appx. weeks 1-6
  - Assignment 2 (30%), due appx. weeks 7-11
  - Milestones due every week from week one (must be submitted by the beginning of your lab)!

- End of Semester Exam
  - Worth 50%
  - Closed Book Exam
Course Resources

- **Course Web Site**
  - via Learning@Griffith and
    http://www.ict.griffith.edu.au/teaching/code
  - Check Notice Board regularly!
  - Read the Policies Page

- **Help outside the Lab**
  - Use Virgil Message Forum
  - Received your Password? – Check official Student EMail!

- **Web Resources**
  - Loads of Online Material via the Web Page!

- **Books, Article, Papers**
  - See the Resources Section!
  - References at the End of each Lecture!
Course Communication

- **Notice Board**
  - Important updates and changes

- **Forum**
  - For Student/Tutor/Lecturer communication
  - Help other students if you can
    - Good feedback for yourself to see how well you have understood a topic!

- **Web Material**
  - Lecture Notes, Articles, Tutorials
  - Code Examples, Model Solutions
  - Made available progressively
    - Check Web Pages regularly
Policy Guidelines

- **Student Policies Web Page**

- **Problems, Consultation, and Grievances**
  - Use the Forum about course-related problems (available any time)!
  - Talk to Lecturer/Tutor at Lectures, Labs, and Tutorials
  - Open Door Policy
    - Drop by my office any time the door is open!
    - EMail me for an appointment at a specified time!
Course Objectives

- Assist in . . .
  - . . . developing correct, efficient, robust, maintainable, and reusable software
- Broaden your programming experience
  - Writing more advanced Programs
- Detailed understanding of . . .
  - . . . data structures, their use and implementation
  - . . . memory and Object management
  - . . . managing complexity
Text Books

- **Recommended Book**

- **No Prescribed Textbook**!
  - Tons of available Web Material
  - Learning how to program is an individual process
    - Making mistakes and learning from them (requires a lot of patience – satisfaction of finally getting it right!)

- **Other Books**
  - Books Section on the course Resources Page
Modules Outline (Preliminary)

- Programming Environment
  - Shell, Compiler
  - Makefiles
  - Subversion Repository
    - How to Submit Milestones?

- Learning New Programming Languages
  - Objective-C (superset of C)
  - C++ (optional)
Modules Outline (Preliminary, continued)

- Making Complex Problems Simple
  - Object Oriented Programming Revisited
  - Computer Architecture Revisited
  - Designing and Managing your own Objects

- Data Structures and Algorithms
  - Collection Frameworks
  - Abstract Data Types and How to Use them
  - Important everyday Algorithms
  - Behind The Scenes implementation of Collection and other Data Structures
Administrativa: That’s It!

- Any Questions?