

Compiler Design: Housekeeping

Housekeeping

1. Lecturer's details
2. Teaching modes
3. IVLE
4. Textbooks
5. Assessment
6. Software

Lecturer's Details

- Name: Razvan Voicu (please call me *Razvan*)
- email: `razvan@comp.nus.edu.sg`
- Google Talk: `razvan.voicu.sg@gmail.com`
- Skype: `razvan.voicu.pda`
- office phone: 6516 2732
- office: COM2-3-10
- Office hours: TBA
 - appointment by email, will always make time
 - usually in during afternoons
 - call office phone before dropping by my office

Teaching Modes: Lectures

- Lecture notes will developed as a wiki
 - allow readers to give feedback and discuss
 - allows prompt corrections if necessary
- No simultaneous webcast
- Screen will be captured and recording will be uploaded to IVLE
- No particular textbook will be followed
 - references to the optional textbooks and various internet resources will be made whenever possible

Teaching Modes: Recitations

- 1 hour classes
 - Exercise set given out in advance
 - Solution developed in class through discussions and collaboration
- Start in week 3
- Screen-capture each session for further revision

Teaching Modes: Homework

- 6 Problem Sets
- Multiple languages
- Many optional components
- Reinforcement of concepts taught in class and also exercises that require creative solutions
- Programming is a **skill** – it requires practice
 - homework exercises will provide plenty of opportunity for that
- Solving homework exercises: **best preparation for exams**

Teaching Modes: Wikied Materials

- A wiki skeleton has been created at:
`cs4212-nus.wikispaces.com`
- Accounts will be created for each student
- Each student will be assigned a set of pages to develop and maintain
 - The page for a topic must be created *after* the lecture on that topic has been held, and the recording is made available
- Everybody is then invited to discuss and suggest improvements
 - The maintainer of each page must continue to maintain the page till the end of the semester.
- Hopefully, full set of notes will emerge; you may take these into the final exam
- Marks will be assigned based on the quality of the contributions

IVLE

- All course information
- Lesson plan
- Announcements and reminders
- General repository for materials
- Homework submission
- Gradebook

Optional Textbooks

- [dragonbook]
Aho et al, "Compilers: Principles, Techniques, and Tools", 2nd ed., ISBN: 0321491696
- [appel]
Appel, A., "Modern Compiler Implementation in C", ISBN:0521586534
- [sterling]
Sterling et al, "The Art of Prolog", ISBN:0262193388

Assessment/Feedback

- Learning component: 50%
 - Overall marks reward effort rather than knowledge
 - no penalty for slow-paced learners
 - Individual marks represent feedback on your performance in each exercise
 - Implemented as a **cap system** (more details on IVLE page)
 - Anybody who puts in a decent level of effort should attain the maximum
- Assessment: 50%
 - Final exam: 50% (open book)
 - Emphasize knowledge, rather than effort

Software

- Pentium assembly language
 - gas (from linux/cygwin binutils)
 - Linux: use "add software" to add the "binutils" package
 - Windows: install cygwin (from cygwin.com), and then the "binutils" package
 - Windows (alternative): install MinGW
- C/C++
 - Install GCC in Linux/Cygwin
 - Windows alternative: Install Visual Studio Express Edition
- Prolog/CLP
 - <http://www.swi-prolog.org/>
 - <http://eclipseclp.org/>

Typical Compiler Topics

- Syntactic analysis
 - Lexical analysis
 - Parsing
 - Abstract Syntax Trees
- Syntax-directed translation
 - Intermediate code
 - Program analysis
 - High-level optimizations
- Code generation
 - Register allocation
 - Instruction selection
 - Low-level optimizations
- Programming Environments
 - Standard libraries
 - Virtual machines
 - Memory management/garbage collection

Our Approach

- Emphasize *Syntax-directed translation*
 - Hands-on approach
 - Prolog as a programming medium
- De-emphasize *Syntactic analysis*
 - Covered via automatic parser and lexer generators
 - Mainly by example
- Cover properly *Code generation*
 - Leave advanced topics such as *low level optimizations* to cs5214
- Discuss issues related to programming environments

Syllabus

- Overview of GCC and Pentium assembly language
- Introduction to Prolog and language manipulation
- Compilation of expressions. Introduction to semantics.
- Compilation of elementary statements.
- Compilation of scopes and procedures.
- Compilation of types, pointers, arrays, and data structures.
- Compilation of object-oriented languages.
- Lexical Analysis and Parsing
- Code generation and optimization.
- Run-time environments.