Note 1

Overture to Programming Methodologies

Kyuseok Shim

http://ee.snu.ac.k/~shim

Associate Professor

Seoul National University

(Class slides are provided by Yunheung Paek)

Topics

- General course information
  - TAs, course web site
  - grading, exams, assignments
  - objectives of this course
  - class schedule
- Introduction to the Scheme language

General information

- Instructor: 심 규 석
  - Office hour: by prior appointment thru e-mail
  - ☎: 880-7269, Email: shim@ee.snu.ac.kr
- TA: 이정훈, 김영훈(880-1758)
  - Email : jhlee@kdd.snu.ac.kr, yhkim@kdd.snu.ac.kr
- References
  - Lecture notes: main reference source
  - Any textbooks similar to these books are OK!
    - R. Sebesta, *Concepts of Programming Languages*, Addison-Wesely
    - R. Sethi, *Programming Languages: Concepts and Constructs*, Addison-Wesley
- Lecture notes will be available online
  - Web page: http://ee.snu.ac.kr/korean/class/class.php?class_no=2491&order4=2

Tentative grading policy

- Two exams:
- Assignments:
  - about many small assignments
  - about 3~4 medium assignments
  - mostly simple programming assignments using three programming languages (C, C++ and Scheme)
  - Details will be given by the TAs later.
- Class attendance:

Lecture information

- Organization of the lecture

  50 ~ 60 min lecture with no break

- Regular lectures
  - 화, 목요일 9:00
  - 1시간 15 분 강의
  - 강의실 : 301-201
- Programming lab hour
  - 금요일
  - 1시간 + α
  - 실습실: 301-207

Objectives of this course

- make it better to understand languages you have been using
- allow better choice of programming languages as an engineer who needs programming to solve his/her problem
- make it easier to learn new programming languages
- increase vocabulary of programming constructs
- write better programs
- make it easier to design/implement a new language (e.g., domain-specific languages in engineering fields)
- ultimately, help you fulfill the course requirements and get a better job.

  → *Every engineer today needs a good programming skill!*

What to cover

- Principles of programming for software development
- Programming language concepts and constructs
  - types, polymorphism, coercion, overloading
  - expressions, assignments, conditional statements
  - procedures/functions, parameter passing
  - blocks, storage managements scope, binding
  - modules, data abstraction, object abstraction
- Theoretical and practical issues
- Programming exercise with different language paradigms
  - imperative programming
  - object-oriented programming
  - parallel programming

Tentative class schedule

구 분	강의내용	실습내용
1 주	Class information	Scheme 실습
2 주	Scheme introduction	
3 주	Software specification	
4 주	Programming language classification	C 실습
5 주	Basic concepts & constructs (Types, Polymorphism)	
6 주	Basic concepts & constructs (Variables, Scopes, Binding)	
7 주	Basic concepts & constructs (Expressions, Statements)	
8 주	Basic concepts & constructs (Functions, Parameter passing)	C++ 실습
9 주	Basic concepts & constructs (Higher-order functions)	
10 주	Basic concepts & constructs (Blocks, Modules)	
11 주	Object-oriented programming	
12 주	Object-oriented programming	
13 주	Additional issues on OOP with C++	MPI 실습
14 주	Parallel programming	
15 주	Parallel programming	

Basic concepts and constructs