CSCE 314
Programming Languages

Introduction and Course Administration

Dr. Hyunyoung Lee

hlee@cse.tamu.edu

414B HRBB

979-845-2490

CSCE 314: Programming Languages

- Course Homepage

  http://faculty.cse.tamu.edu/hlee/csce314/

- Textbooks

  1. *Programming in Haskell, 2nd Ed.*, Graham Hutton, Cambridge University Press
  2. *The Java Programming Language, 4th Ed,* Ken Arnold, James Gosling, and David Holmes, Addison-Wesley Professional

- Other Reading Material: See the course homepage

CSCE 314: Programming Languages

- Prerequisites:
  - CSCE 221 (or concurrent enrollment)
  - CSCE 222 (not official, but implicit)
  - Familiarity with recursion
  - Familiarity with complexity analysis
  - Eager to learn (enthusiasm)
  - Perseverance
  - Positive, constructive thinking
  - …

Motivations

- Ever increasing complexity and the size of modern computer programs

- Ever increasing time and cost of program development

- Ever increasing requirements on the correctness of the programs

- *Ever increasing need for a better programmer who learns new languages quickly!*

Course Objectives (1)

Understand the fundamental concepts of the programming languages:

- Broader understanding of language constructs, common abstraction mechanisms, and efficiency consideration

- Understanding of the basics of how programs written in high-level programming languages are executed, i.e., parsing, internal program representation, type checking, interpretation

Course Objectives (2)

Understand the fundamental concepts of the programming languages:

- Encourage the use of formal verification
- Effective use of the functional programming approach to design and implement programs
- Understanding of the main features of modern object-oriented languages

Course Outline (1)

- Language Processing
  - Grammars, lexing and parsing
  - Abstract syntax, internal representations
  - Types and type checking
  - Interpretation
  - Basics of an implementation of a simple programming language

Course Outline (2)

- Study of a Functional Language - Haskell
  - Type inference
  - Parametric polymorphism
  - Higher-order functions
  - Algebraic data types
  - Abstract data types and modules
  - Type classes
  - Effects in a “pure” language

Course Outline (3)

- Study of an Object-Oriented Language - Java
  - Subtyping and inheritance, subtype polymorphism
  - Exception handling
  - Generics, wildcards
  - Reflection
  - Concurrency

Grading

- Homework: 25%
- Midterm Examination: 30%
- Final Examination: 30%
- Quizzes: 10%
- Exercises: 5%

Assignments and Submission

- Homework will be assigned approx. bi-weekly
- Turn in all work on eCampus before the deadline
- All assignments will be done individually and will include programming, essay-style problems, and problems from the textbook. It is extremely important to work on the homework assignments carefully!

Assignments (Cont.)

- Late turn-in will have penalty @2% off per hour late
- Genuine difficulties must be discussed with Instructor *before the deadline*
- Never cut classes to do your homework
- Some class time will be devoted to hands-on learning (exercises, discussion on homework, etc.)

Collaboration Policy

- Acceptable collaboration includes: (1) discussing the assigned problems to better understand their meaning or (2) discussing possible approaches to assigned problems. Explicitly acknowledge any help received from someone and reference every source you use, whether it is a person, a book, a paper, a solution set, a web page or whatever.
- Unacceptable collaboration includes: (1) copying (verbatim use) of physical papers or computer files (including program files), (2) submission of solutions that are jointly authored, or authored either wholly or in part by other individual, or (3) providing physical papers or computer files (including program files) of your (or third-party) solutions to other individuals.

Earning grade in this course

- Planning to get a good grade?

  Attend all classes, pay attention in the class, do exercises, do assignments, read book and reading materials, participate in the class

- How to get a bad (non-passing) grade?

  Miss classes, do not pay attention in the class, do not do assignments, do not read reading materials, . . .

Summary

- Read syllabus on how to conduct in the course
- We will study fundamentals of programming languages by way of learning two languages – Haskell and Java
- A lot of “fun” work in the class
- Happy learning!!