

ECE326

PROGRAMMING LANGUAGES

Lecture 1 : Course Introduction

Kuei (Jack) Sun

ECE

University of Toronto

Fall 2019

Course Instructor

- Kuei (Jack) Sun
- Contact Information
 - Use Piazza
 - Send me (Kuei Sun) a private post
 - Not “Instructors”, otherwise the TAs will also see it
 - <http://piazza.com/utoronto.ca/fall2019/ece326>
 - Sign up to the course to get access
 - Office: PRA371 (office hours upon request only)
- Research Interests
 - Systems programming, software optimization, etc.


Course Information

- Course Website
 - <http://fs.csl.toronto.edu/~sunk/ece326.html>
 - Lecture notes, tutorial slides, assignment handouts
- Quercus
 - Grade posting, course evaluation, lab group sign-up!
- Piazza
 - Course announcement, course discussion
 - Assignment discussion
 - Lab TAs will read and answer relevant posts periodically

Course Information

- No required textbook
 - Exam questions will come from lectures, tutorials, and assignments
 - See course website for suggested textbooks
- Lab sessions
 - Get face-to-face help from a TA with your assignments
- Tutorials
 - Cover supplementary materials not in lectures
 - Go through sample problems that *may* appear on exams

Volunteer Notetakers

- Help your peers
- Improve your own notetaking skills
- Receive Certificate of Recognition
- Information:
 - <https://www.studentlife.utoronto.ca/as/note-taking>
- Registration:
 - <https://clockwork.studentlife.utoronto.ca/custom/misc/home.aspx>
- Notetakers needed for lectures and tutorials 

Background

- Programming Languages
 - A formal language consisting of a instructions to implement algorithms and perform tasks on computers
 - Thousands exist, more being made
- Programming Paradigms
 - A way to categorize programming languages by features
 - Execution model
 - Code organization
 - Style and syntax
 - ...etc

Course Outline

- Imperative Programming
 - Writing instructions and commands (APS105)
- Object-Oriented Programming
 - Objects interacting with one another (ECE244)
- Metaprogramming
 - Writing code that generates more code (CSC324)
- Concurrent Programming
 - Multiple tasks overlapping in their executions (ECE344, CSC367)
- Functional Programming
 - Programming in the style of evaluating mathematical functions (CSC324)

Course Objective

- Learn different ways of writing computer programs
 - Law of the instrument
 - “To the man who only has a hammer, everything he encounters begins to look like a nail.” – Abraham Maslow
 - One style may work better than others for a given problem
 - E.g. recursion vs. iteration
- Analyze programming languages and their features
 - Semantic: more expressive power
 - Syntactic: easier to read/write
 - Optimization: improves performance and efficiency

Syntactic Sugar

- Example: Java

- For Loop

```
String[] fruits = { "Apple", "Banana", "Strawberry" };  
for (int i = 0; i < fruits.length; i++) {  
    system.out.println(fruits[i]);  
}
```

- For-Each Loop

```
for (String f : fruits) {  
    system.out.println(f);  
}
```

- Same bytecode generated, but easier to read

Compiler Optimization

- Example: C

- restrict keyword

- Informs compiler that a pointer has no alias

```
void add2(int * a, int * b, int * restrict c) {  
    *a += *c;  
    // normally, *c must be reloaded because a may  
    // point to same address as c, which means that  
    // *c could change after "*a += *c" is executed  
    *b += *c;  
}
```

- Compiler can thus be more aggressive in optimizing this function

Course Objective

- Learn implementations of language features
 - Cost-benefit analysis
 - E.g. automatic memory management
 - Avoids bugs associated with manual memory management
 - Requires garbage collection
- Learn programming languages *that matters*
 - Popularity – large community of active developers
 - Easy to find help
 - Easy to find libraries or packages
 - Easy to find jobs

Popularity

- By job posting
 - Java: 65,986 jobs
 - Mainly used by web application developers
 - Python: 61,818 jobs
 - Known for high productivity
 - JavaScript: 38,018 jobs
 - The de facto language for modern browsers
 - C++: 36,798 jobs
 - Favored by game developers, large application software, ...etc.
 - C#, PHP, Perl...

Source: <https://www.codingdojo.com/blog/the-7-most-in-demand-programming-languages-of-2019>

Popularity

- By contribution on GitHub
 - JavaScript
 - Java
 - Python
 - PHP
 - C++
 - C#, TypeScript, Shell, C, Ruby, ...etc
- Correlates well with popularity by job postings

Source: <https://github.blog/2018-11-15-state-of-the-octoverse-top-programming-languages/>

Popularity

- By Google search trends
 - JavaScript
 - Python
 - Java
 - Go
 - Google's programming language
 - Aims to simplify programming in multicore, networking environment
 - Elixir
 - Ruby, Kotlin, TypeScript, Scala, Clojure, ..etc.

Source: <https://codeburst.io/10-top-programming-languages-in-2019-for-developers-a2921798d652>

Popularity

- By growth in community
 - Kotlin
 - HCL
 - TypeScript
 - PowerShell
 - **Rust**
 - CMake
 - Go
 - Python, Groovy, SQLPL

Source: <https://github.blog/2018-11-15-state-of-the-octoverse-top-programming-languages/>

Lab Assignments

- 4 Assignments
- 3 Programming languages
 - C++11
 - A newer version of C++ compared to what's learned in ECE244
 - Python 3
 - Extremely popular high-level programming language
 - Rust
 - Fairly new systems programming language
 - Focuses on performance and safety
 - Requires you to *really* understand your own code

Lab Assignments

1. Easy Blackjack
 - Play the game automatically
2. Optimal strategy calculator
 - Play Blackjack optimally
3. Object-relational mapping
 - Seamless integration of objects with a relational database
4. Concurrent Database
 - Implement a fast toy database

Lab Assignments

- Expect significant efforts involved
 - You *will* need to spend time outside of lab hours
 - Learn new programming languages
 - Apply new programming techniques
 - Require time to design and implement your program
 - Require *lots* of time for debugging and performance improvement
- Groups of 2
 - Sign up for a group on Quercus
 - Required for you to gain access to Git repository
 - Allows you to work on a common code base
 - Also used to submit your assignment

Lab Information

- You may go to any of the lab sessions for TA help
 - Lab times and location posted on course website
- After hour support (on Piazza)
 - Your peers!
 - Teaching Assistants
 - Assignment 1: Xue (Chloe) Geng
 - Assignment 2: Wenjun (Wendy) Qiu
 - Assignment 3: Szu-Chieh (Jeffrey) Fang
 - Assignment 4: Jemin Andrew Choi
 - Me
 - I won't answer unless the problem is with the assignment

Assignment Grading

- Automated Tester
 - Runs a set of tests
 - Correctness
 - Performance
 - Gives you a mark on your solution
 - WYSIWYG
 - Mark you get from tester is what you expect for the assignment
- Assignment Submission
 - Follow lab instructions
 - You may submit multiple times – do not submit last minute!

Assignment Policy

- No deadline extensions
 - Failure to submit on time will result in zero marks
 - There's two of you per group
 - You are given 3 to 5 weeks per assignment
 - *start early, don't procrastinate!*
 - Make sure you submit something for partial marks
- Plagiarism and Cheating
 - Severe penalty
 - Grade reduction to suspension from University
 - Talk to me if you feel overwhelmed

Grading Scheme

- Midterm: 25%
 - October 29th, 4pm to 5:30pm, EXE320
- Final: 45%
- Assignments: 26%
 - 5%, 6%, 7%, 8%, respectively
- Tutorial Quizzes: 4%
 - 12 tutorials, 12 quizzes
 - Only need to complete 10/12 quizzes for full marks

Next Lecture

- “Classifications of Programming Languages”
- Instructor: Kuei (Jack) Sun
- Course Website:
<http://fs.csl.toronto.edu/~sunk/ece326.html>
- Piazza Discussion:
<http://piazza.com/utoronto.ca/fall2019/ece326>