Basis and Practice in Programming
프로그래밍 기초와 실습

Sun Jae Lee
이선재

Dept of Computer Science and Engineering,
Sungkyunkwan University (SKKU)

About Me

- Sunjae Lee (이선재)
  - Education
    - PhD in CS from KAIST, 2025
    - MS in CS from KAIST, 2021
    - BS in CS from KAIST, 2019
  - Professional Experiences
    - **Assistant Professor**, Department of Computer Science and Engineering, SKKU, Mar. 2025 - present
    - **Chief Technology Officer (CTO)**, Fluiz, Jan. 2022 – Feb. 2025
      - » https://fluiz.ai/
  - Research interests: **Mobile-AI systems**
    - Multi-device Mobile Platform
    - Mobile Operating Systems
    - Mobile AI Assistant
    - AI Agent systems

About Me

- Contact
  - Office: Engineering 2 #27324 (제 2공학관 27동 3층 27324호)
  - Homepage: https://sunjae1294.github.io/
  - Email: sunjae.lee@skku.edu
  - Office hours: by appointments

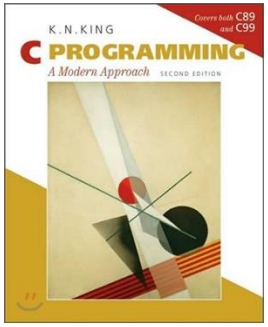
Welcome to Basis in Programming!

- **Class schedule**
  - Lecture session: Tuesday (12:00 – 13:15)
  - Practice session: Thursday (13:30 – 14:45)
    - May include little bit of lecture
  - Place: Chemistry #330110 (화학관 첨단강의실 [330110])

- **Attendance**
  - via Electronic Attendance System
    - https://attend.skku.edu
  - Present: -10 Min ~ Class start time ~ +15 min
  - Late: Class start time + 15 min ~ 30 min
  - Absent: Class start time + 30min or absent
  - **For Practice session, you MUST bring your laptop. If you don't you will be marked absent.**
  - **You will be given F if you are absent 8 times or more.**

Welcome to Basis in Programming!

- **Class materials**
  - **https://icampus.skku.edu/**
    - Lecture videos
    - Lecture notes
      - » Adapted from slides of Prof. Sangeun Oh (Korea University), Prof. Jinkyu Lee (SKKU), and Prof. Young Hoon Kim (SKKU)
    - Important notices
    - Q&A board
      - » If you have private questions, ask me via email: sunjae.lee@skku.edu
  - **https://skku.goorm.io/**
    - Registration code: To be announced
    - Practice session
    - Coding practice
  - Book
    - Most of the materials are on the slides. But if you wish to buy a book:
    - "C Programming : A Modern Approach, 2nd edition", by K. N. King, W. W. Norton & Company



Welcome to Basis in Programming!

- **Important dates:**
  - **05.05:** Holiday (어린이날 대체휴일)
  - **04.24:** Midterm Exam
  - **06.19:** Final Exam
  - If you can't take exam on these dates, please email me (sunjae.lee@skku.edu)
- **Grades:**
  - 10%: Attendance
  - 20%: Practice session (participation, assignment, …)
  - 35%: Midterm Exam
  - 35%: Final Exam
  - If you don't show up on exam, you will automatically fail this course

Welcome to Basis in Programming!

- **Exams:**
  - Location: You will take exams in class using your laptop (on-site online exam)
  - Coverage: Everything (Lecture + Lab) taught in this course
    - Final exam will cover all course contents.
  - More detailed instructions will be provided before each exam.
- **Practice session:**
  - **Before midterm:** You will solve coding problems using Goorm.
  - **After midterm:** You will work on a coding project to make a simple 2D game.
  - More information on Thursday's practice session.
- **Assignments:**
  - Any tasks not completed during the practice session will become your assignments
  - You must finish them before coming to the next practice session.

Welcome to Basis in Programming!

- **Cheating policy:**
  - **Zero-tolerance policy (No mercy!)**
  - Any attempt to cheat will result in an F.
  - Examples of cheating
    - Sending your codes to others
    - Copying other's code
    - Fabricating excuses for attendance
    - Using ChatGPT (or any other AI tools) for assignment or during practice session
    - Any other form of dishonesty
  - We encourage discussion among students, but never share your source code.
    - If you are caught copying code, both the giver and the receiver will be penalized.

Welcome to Basis in Programming!

- **ChatGPT (or any other AI tools) policy:**
  - **Coding: Do not use it for coding.**
    - Yes, it is convenient, but it is for your own good.
    - If you are **really really stuck** on your coding assignment, ask for my permission to use ChatGPT in advance.
  - **Lecture Content:** You may use it to clarify concepts or summarize lecture materials.
    - But, asking questions and participating in discussion is strongly encouraged.

Lecture Overview

- **Week 1**
  - Course introduction and Programming Basics
- **Week 2-7**
  - Programming Basics
  - including data types, flow control, functions
- **Week 8**
  - Midterm exam
- **Week 9-15**
  - Intermediate Programming
  - including pointers, data structures, low-level programming
- **Week 16**
  - Final exam

What is programming?

https://www.youtube.com/watch?v=vYuER_3og_M

Then Why Study *Programming*?

https://www.youtube.com/watch?v=qYZF6oIZtfc

What is programming *"Language"?*

Computer hardware

Machine Code

```
10011101000110100000
01100011010001110110
10000010111101101110
11110110001011011000
10000010011100011011
10010011000111000000
```

Machine code (binary code)

```
push    ebp
mov     ebp, esp
movzx   ecx, [ebp+arg_0]
pop     ebp
movzx   dx, cl
lea     eax, [edx+edx]
add     eax, edx
shl     eax, 2
```

Assembly language

What is programming *"Language"*?

```
push    ebp
mov     ebp, esp
movzx   ecx, [ebp+arg_0]
pop     ebp
movzx   dx, cl
lea     eax, [edx+edx]
add     eax, edx
shl     eax, 2
```

Assembly language

```
#include <stdio.h>

int main() {

    printf("Hello, World!\n");

    return 0;
}
```

C language (High-level language)

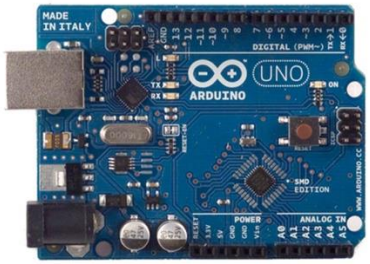
➔ Still need to understand H/W details

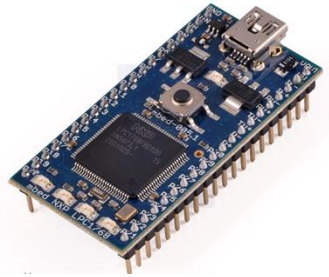
Other high-level languages

➔ Hiding H/W details for the easiness of programming

Why Studying "*C*" *Language*?

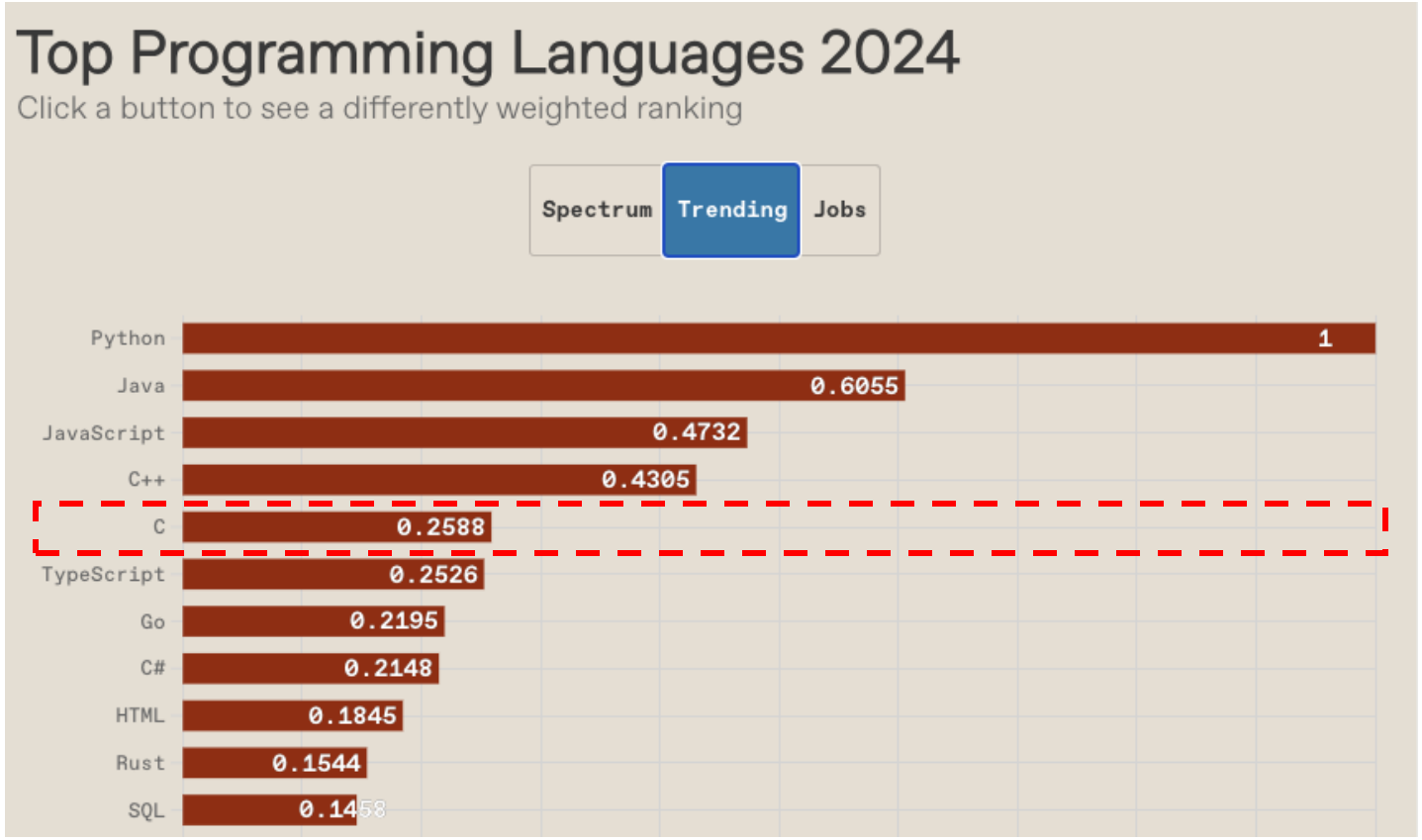
- Computer science majors must understand not only *high-level programming concept* but also *low-level hardware details*!
  - C language is a powerful programming language that lets you understand low-level hardware interactions.
- A significant portion of core computer systems are built with C:
  - Operating systems (e.g., Windows, Linux, Android)
  - Compilers, Linkers, Loaders, etc.
  - Embedded systems





Why Studying "*C*" *Language*?

- Still, one of the most widely used language
  - https://spectrum.ieee.org/top-programming-languages-2024



How programming works

- **Making goals**
  - Understanding requirements in given problems
- **Writing algorithms**
  - Designing flow / logic of your program
- **Coding**
  - Translating the algorithm into programming language
- **Testing**
  - Test whether correct outcomes are obtained
- **Debugging**
  - Modifying code to correct errors found during testing.

How programming works

- **Making goals**
  - Understanding requirements in given problems
- **Writing algorithms**
  - Designing flow / logic of your program

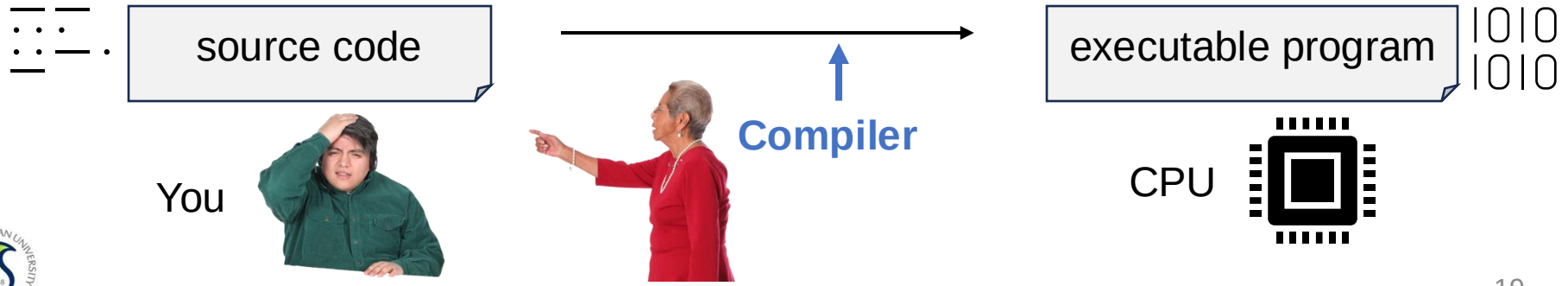
Hardest part!!

- **Coding**
  - Translating the algorithm into programming language → Easy!
- **Testing**
  - Test whether correct outcomes are obtained
- **Debugging**
  - Modifying code to correct errors found during testing.

Where you will spend most time

How programming works

- **Code Editor**
  - Where you write your source code.
  - Text editor (e.g., Word, 한글, notepad) for programming languages
  - Checks your code's *grammar* (syntax) real-times
- **Compiler**
  - Converts your source code into a machine-understandable binary code (executable).
  - Checks if your code *makes sense*.
  - Comes with programming language. You don't have to worry about it



Tips and advise for computer science students

- ***Don't be a coder, be a computer scientist (or software engineer)***
  - Coders build what they're told. Computer scientists design what *should* be built.
  - Anyone can learn to code, but not everyone can be a computer scientist
    - Anyone can learn to write, but not everyone can write a novel.
- ***Programming language is just a tool, not the goal***
  - You must be able to choose the right tool for different problems.
  - Familiarize yourself with variety of tools: C++, Java, python, Rust, …
- ***Think first, code second***
  - Spend more time thinking, less time coding.
  - This will eventually make you better, faster programmer.
- ***Ultimately, it is all about problem solving***
  - Problems change, tools change, but your ability to solve problems remains