Lec 1: Introduction

CS220: Programming Principles

Sang Kil Cha
Who am I?
Sang Kil Cha

• Researcher and Software Engineer.
Sang Kil Cha

- Researcher and Software Engineer.
- Leader of SoftSec. Lab.
Sang Kil Cha

• Researcher and Software Engineer.
• Leader of SoftSec. Lab.
• Director of CSRC (Cyber Security Research Center)
• Chief professor of GSIS.
Sang Kil Cha

• Researcher and Software Engineer.
• Leader of SoftSec. Lab.
• Director of CSRC (Cyber Security Research Center)
• Chief professor of GSIS.
• Research Keywords:
  - Binary Analysis
  - Vulnerability Discovery
  - Exploit Verification
  - Malware Analysis
Contact

• Office: N5, #2319
• Office hour: by appointment.
  - 🏠 Homepage: https://softsec.kaist.ac.kr/~sangkilc/
  - 📧 Email: sangkilc@kaist.ac.kr
My Research
My research is all about building large and complex systems that automatically analyze programs to resolve security problems.
My Research: Automatic Exploit Generation

Fully automated hacking and defense (e.g., DARPA’s Cyber Grand Challenge)
Normal CTFs

Team A ← Team B

Team C
Normal CTFs

Team A

Team B

Team C

Vulnerable App

Vulnerable App

Vulnerable App
Normal CTFs

Team A

Vulnerable App

Team B

Vulnerable App

Team C

Vulnerable App
Winner = Mayhem

ForAllSecure (Carnegie Mellon University)

Winner = Mayhem

ForAllSecure (Carnegie Mellon University)

2012 IEEE Symposium on Security and Privacy

Unleashing MAYHEM on Binary Code

Sang Kil Cha, Thanassis Avgerinos, Alexandre Rebert and David Brumley
Carnegie Mellon University
Pittsburgh, PA
{sangkilc, thanassis, alexandre.rebert, dbrumley}@cmu.edu

IEEE S&P Test-of-Time Award 2022

IEEE Symposium on Security and Privacy

Test-of-Time Award

Unleashing Mayhem on Binary Code

Sang Kil Cha, Thanassis Avgerinos, Alexandre Rebert, and David Brumley (2012)

San Francisco, USA, May 2022
My Research: Windows Error Reporting

Image from https://goo.gl/PLekeyZ
My Research: Windows Kernel Vulnerabilities

- Published in 2021 IEEE Security & Privacy.

- Earned $25,000 USD bounty for finding zero days.
My Research: Automatic Bug Finding

• Targeting **various software products**: browsers, kernels, smart contracts, etc.
• Practical impact: numerous CVEs, algorithms used by mainstream fuzzers, etc.
• Academic impact
  - 2020 FSE Distinguished Paper Award.
  - 2022 TSE Best Paper Award.
  - 2022 ASE Distinguished Paper Award.
My Research: Binary Analysis

Idea

Source Code

Intermediate Representation

Assembly

Binary Code

Compile

Reversing
My Research: B2R2

• The fastest binary analysis frontend.
• Academic impact
  - 2019 NDSS BAR Best Paper Award.
My Research: Online Game Security

• Automatic aimbot detection.
• Academic impact
  - 2023 USENIX Security Distinguished Paper Award.
Q: But how were all these possible?
Q: But how were all these possible?

A: Right language and right engineering principles.
You will learn fundamental *principles* for programming, and how to apply such concepts to build real-world programs using a state-of-the-art and practical language called *F#*. 
Q: Is F# used in practice?

A: Yes!
An Anecdote: MSRD

The engineering team, consisting of three developers at the time, was given the ambitious goal to build an entirely new service from scratch and ship it to external customers in just three months\(^1\).

\(^1\)See: https://goo.gl/CJvmdq
Q: How about Salary?
What Languages Are Associated with the Highest Salaries Worldwide?

<table>
<thead>
<tr>
<th>Language</th>
<th>Salary (in $)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F#</td>
<td>$74,000</td>
</tr>
<tr>
<td>Ocaml</td>
<td>$73,000</td>
</tr>
<tr>
<td>Clojure</td>
<td>$72,000</td>
</tr>
<tr>
<td>Groovy</td>
<td>$72,000</td>
</tr>
<tr>
<td>Perl</td>
<td>$69,000</td>
</tr>
<tr>
<td>Rust</td>
<td>$69,000</td>
</tr>
<tr>
<td>Erlang</td>
<td>$67,000</td>
</tr>
</tbody>
</table>

¹ Developer survey result in 2018 by stackoverflow.com.
How about in 2023?

1. Zig
2. Erlang
3. F#
4. Ruby
5. Clojure
6. Elixir
7. Lisp
8. Scala
9. ...

---

¹See https://survey.stackoverflow.co/2023/
Q: Most admired S/W framework/library by developers?
1. .NET
2. Numpy
3. TensorFlow
4. Pandas
5. Torch
6. Flutter
7. Apache Kafka
8. ...

¹See https://survey.stackoverflow.co/2023/
Course Overview

- **Approach:** learn fundamental programming principles.
- **Target Audience:**
  - One who has basic understanding of programming.
  - Students who have *taken CS101*, or have equivalent knowledge/skills.
The Most Important Thing: Academic Integrity

Any solution you submit (quiz, exam, etc.) must be your own work.

No cheating, no plagiarism.

If you violate this rule, you will immediately get an F, and possibly get expelled from the university.
Honor Code

To take this course, you **must** sign the Google form.
Course Attendance

We will check your course attendance in the middle of each class.

- Check every class.
- Being late means you are absent.
- Three days of freedom: you can safely be absent for up to three lectures without affecting your grade.
- If you miss four or more classes, you will get an ‘F’.

No excuse, no compromise.
I understand, but what if I have a personal occasion?

That is exactly why we give the three-day of freedom. Use your time *wisely*.

Q: What if I really have a serious problem, e.g., a significant health problem, etc.? Can this be an excuse?

A: We accept *only* official documents (e.g., medical certificate).
Our Agreement

During the course, if you did not attend 89% of the lectures, you will get an ‘F’ no matter what reasons you may have.
Questions in Korean

I love questions. You can use **Korean** to ask questions during the class. I will translate your question into English for other students.
Course Resources

- **Asking Questions:** Use GitHub discussions. You will see the Q&A category.
  - [https://github.com/KAIST-CS220/Main](https://github.com/KAIST-CS220/Main)
  - *English only.* If needed, use ChatGPT or similar to translate your question.
  - We do **not** accept (will ignore) emails asking questions about homework.

- **Syllabus & Slides:**
  [https://softsec.kaist.ac.kr/courses/2024s-cs220/](https://softsec.kaist.ac.kr/courses/2024s-cs220/)
Emails

- Instructors have right to ignore your personal email especially when it is about homework. Use the GitHub discussions instead.
- We do accept emails for *administrative* issues only, e.g., when you were not able attend the midterm exam due to a critical reason.
Text Book

No official text book, but this course is inspired by the following great books:

- The Wizard book².
- Essential F#.
- Functional Programming Using F#.

²“Structure and Interpretation of Computer Programs (2nd Edition)” by Harold Abelson and Gerald Jay Sussman.
Course Logistics

• 5% Participation (attendance, in-class activities, etc.).
• 20% Homework
• 35% Midterm
• 40% Final
Bring Your Own Laptop

We will do in-class coding activities almost every class. You should bring your own laptop to the class.

In-class activities are *not* graded, though.
GitHub Discussion and Participation

• We will use GitHub discussions for Q&A.
• We will check your participation in the discussion.
• If you answer other students’ questions, and if the answer is marked as an answer, you will get a participation point.
• Be sure to mark someone’s comment as an answer if it helps you resolve your question.
I’ve taken CS320 (Programming Language). Can I still take this course?

We recommend you to take this course first, and then take CS320. However, this course has its own value, so it is totally up to you.
Development Environment
Can I use ChatGPT?

Feel free to use ChatGPT or similar for your homework, but midterm and final exams are paper-based and you are not allowed to use any electronic devices. So I recommend you to not use ChatGPT for your homework so that you can practice your programming skills.
Development Environment?

You are free to use any OS you want including Linux, macOS, and Windows, because we are going to use OS-independent tools and language. Regardless of the OS, we assume that you are working on a terminal (a.k.a. console or shell) with a proper editor, such as Emacs, Vim, Visual Studio Code, etc.

Recommended editors:

- Visual Studio
- VSCode (https://code.visualstudio.com/)
Environmental Setup

1. Install latest .NET 8 on your machine. You should be able to run the command ‘dotnet’ in your terminal.

2. Make sure you install .NET 8. If you already have a lower version of .NET installed on your machine, you should remove it first, and then install a latest version.

3. Install Git on your machine. You should be able to run the command ‘git’ in your terminal.
Check Your Installation

If you see something similar to the following, you are good.

On Windows (from cmd.exe)

C:\Users\sangkilc> dotnet --version
8.0.100

C:\Users\sangkilc> git --version
git version 2.42.0.windows.2
Check Your F# REPL

- Open a terminal (e.g., cmd.exe on Windows).
- Type `dotnet fsi` in the prompt.
Check Your F# REPL (cont’d)

$ dotnet fsi

Microsoft (R) F# Interactive version 12.8.0.0 for F# 8.0
Copyright (c) Microsoft Corporation. All Rights Reserved.

For help type #help;;

>
Conclusion
Notes

• From the next class, we will assume that you have successfully configured your development environment.
• Check our course web site, and read references.
• Not-graded homework:
  - Install .NET 8 on your machine.
  - Install Git on your machine.
  - Make sure you can successfully run `dotnet -version` and `git -version` in a terminal.
Question?