

# Lecture 0 – Introduction

## SWS121: Secure Programming

Jihyeok Park



2024 Spring

- **Instructor:** Jihyeok Park (박지혁)
  - **Position:** Assistant Professor in CS, Korea University
  - **Expertise:** Programming Languages, Software Analysis
  - **Office hours:** 14:00–16:00, Tuesdays (appointment by e-mail)
  - **Office:** 609A, Science Library Bldg
  - **Email:** jihyeok\_park@korea.ac.kr
  
- **Class:** SWS121: Secure Programming
  
- **Lectures:** 18:45–20:15, Mon. @ 304 Aegineung (애기능생활관)
  
- **Homepage:** <https://plrg.korea.ac.kr/courses/sws121/>

| <b>Week</b> | <b>Date</b> | <b>Contents</b>              |
|-------------|-------------|------------------------------|
| 1           | 03/04       | Introduction                 |
| 2           | 03/11       | Basics                       |
| 3           | 03/18       | Testing and Documentation    |
| 4           | 03/25       | Classes, Traits, and Objects |
| 5           | 04/01       | Functional Programming       |
| 6           | 04/08       | Immutable Collections        |
| 7           | 04/15       | For Comprehensions           |
| 8           | 04/22       | Midterm Exam Week (No Class) |
| 9           | 04/29       | Lazy Evaluation              |
| 10          | 05/06       | Generics                     |
| 11          | 05/13       | Advanced Types               |
| 12          | 05/20       | Contextual Abstraction       |
| 13          | 05/27       | Metaprogramming              |
| 14          | 06/03       | Concurrent Programming       |
| 15          | 06/10       | Course Review                |
| 16          | 06/17       | Final Exam Week (No Class)   |

- **Homework Assignments: 90%**
  - **3 Programming Assignments:**
    - Homework 1: 30% (due on April 15)
    - Homework 2: 30% (due on May 20)
    - Homework 3: 30% (due on June 17)
  - Submit your homework on [Blackboard](#).
  - You can utilize or refer to any other materials (e.g., ChatGPT), but you **MUST** write your **OWN** solution.
  - **Cheating is strictly prohibited. Cheating will get you an F.**
- **Attendance: 10%**
  - Please use [Blackboard](#) to attend the class **by yourself**.

- **Self-contained lecture notes.**

<https://plrg.korea.ac.kr/courses/sws121/>

- **Reference**

- **“Tour of Scala”**

[docs.scala-lang.org/tour/tour-of-scala.html](https://docs.scala-lang.org/tour/tour-of-scala.html)

- **“Scala 3 Book”**

[docs.scala-lang.org/scala3/book/introduction.html](https://docs.scala-lang.org/scala3/book/introduction.html)

- **“Scala 3 Reference”**

[docs.scala-lang.org/scala3/reference/index.html](https://docs.scala-lang.org/scala3/reference/index.html)

Unexpected faults in **safety-critical software** cause serious problems:

|  |   |   |   |
|--|---|---|---|
| <p><b>June 4, 1996: Ariane-5 explodes after lift off</b></p> <p>Today In History: June 4, 1996: Ariane-5 explodes after lift off</p> <p>Original source: 2008-04-08<br/>Ariane 5, head of archive</p>  | <p><b>Knight Capital Says Trading Glitch Cost It</b></p> <p>BY NATHANIEL POPPER AUGUST 2, 2013 6:07 AM 750</p> <p>Runaway Trades Spread Turmoil Across Wall St.</p>  | <p><b>Heathrow Airport apologises for IT failure disruption</b></p> <p>14 February 2020</p>  | <p><b>Cruise recalls all its driverless cars</b></p> <p>Hit another setback, Cruise updates software on 250 driverless cars to fix its 'Collision Data'</p> <p>By Steve Bannor</p> <p>Updated November 6, 2023 at 2:13 a.m. EST · Published November 6, 2023 at 2:09 a.m. EST</p>  |
| <p><b>Rocket</b></p>   | <p><b>Financial</b></p>   | <p><b>Airport</b></p>   | <p><b>Auto. Vehicle</b></p>   |
| <p>(1996)</p>  | <p>(2012)</p>   | <p>(2020)</p>   | <p>(2023)</p>   |

Then, how can we **prevent** such software faults?

Let's learn **secure programming** to write **safe** and **reliable** software with **Scala**.

**Secure Programming** is a coding practice that ensures the software is designed to be secure and free from vulnerabilities.

- **Static type checking**
  - Using the type system to catch bugs
- **Test-driven development (TDD)**
  - Writing tests before writing the code
- **Documentation**
  - Writing clear and concise comments
- **Encapsulation**
  - Hiding the implementation details
- **Defensive programming**
  - Writing code to handle unexpected inputs



Scala stands for **Scalable Language**.

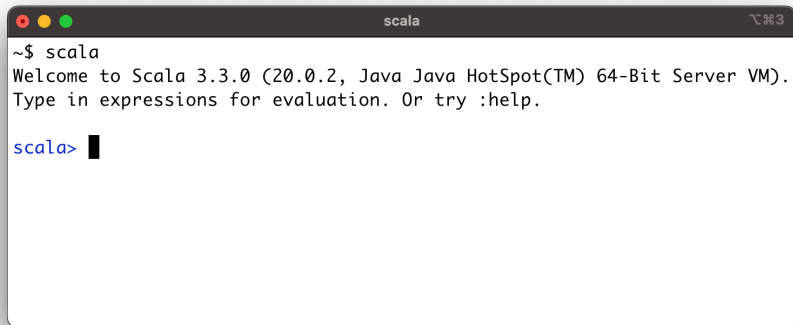
- A **more concise** version of Java with **advanced features**
- A general-purpose programming language
- **Java Virtual Machine (JVM)**-based language
- A **statically typed** language
- An **object-oriented programming (OOP)** language
- A **functional programming (FP)** language



# Read-Eval-Print-Loop (REPL)

Please download and install them using the following links.

- **JDK** – <https://www.oracle.com/java/technologies/downloads/>
- **sbt** – <https://www.scala-sbt.org/download.html>
- **Scala REPL** – <https://www.scala-lang.org/download/>



```
scala
~$ scala
Welcome to Scala 3.3.0 (20.0.2, Java Java HotSpot(TM) 64-Bit Server VM).
Type in expressions for evaluation. Or try :help.

scala> █
```

- Basics

Jihyeok Park  
jihyeok\_park@korea.ac.kr  
<https://plrg.korea.ac.kr>