Programming Fundamentals 2

Pierre Talbot

16 February 2021

University of Luxembourg

UNIVERSITÉ DU
LUXEMBOURG

Chapter 0. Getting Started

What is it?

This class will be about **programming** in Java.

Some aspects of this class are experimental.

- No distinction between lectures and labs.
- Intensive first half: **70% of your grade in 2 months.**
- Feedback on what you produce (quick grading, code review, ...).
- Standard and competitive tracks.

Don't hesitate to help us to improve this class!

Organization

FULL REMOTE: Every Tuesday and Thursday, 8:00 to 9:30.

There is no difference between lectures and labs!

Class layout:

1. **Chapters**: The core notions of Java are divided into 15 chapters.
2. **Live coding**: You watch me coding something.
3. **Code analysis**: We look at your projects and review them.
4. **Crafting**: Learn how to use your tools!

Ezhilmathi Krishnasamy (aka. Mathi) is the TA of this class, he will take a good look at your code and discuss it during code analysis session.

Planning

Two tracks: standard track and competitive track.

Standard track

- 16/02–15/04: **4 labs**, 1 every two weeks (40% of your grade).
- 15/04 (14:00–17:00): **Exam** (30% of your grade).
- 15/04–16/05 (labs 5 and 6): You will fight in the **A.I. Arena** (30% of your grade).
- Beware: coding exam plus **oral exam** for redoing students (100%).

Labs

Three parts: basic exercises + main topic + competitive exercises.

- Lab 1: Connect Four
- Lab 2: Pokedeck
- Lab 3: ?
- Lab 4: Musical Improvisation

BASIC
Pikachu
HP60
Thundershock 10
Flip a coin. If heads, the Defending Pokémon is now Paralyzed.
Tail Whap 20
×2

A.I. Arena

The remaining 30% will be gained by designing an artificial intelligence for a simplified version of a MOBA-like game.

You'll compete against each other for the throne!

Competitive track

- Track unlocked after you complete the standard track.
- Selected competitive exercices.
- You collect additional points.
- **Special events**: Hash code, Google Code Jam, ... (bonus points).

Competitive team

If you are interested, we can set up a team for ACM-ICPC in 1 or 2 years (need more or less preparation depending on your goal).

Competitive track

Coding competitions are very fun, and you learn a lot of new algorithms!

acm International Collegiate Programming Contest

Hash Code

Prologin

Competitive track planning

- 16th February → 15th April: Some **UVa problems** for each lab.
- **Team Event 1: Hash Code**: 25th February, 18:30
  `https://hashcodejudge.withgoogle.com`
- **Event 2: Google Code Jam Qualification**: 26th March, 23:00 to 28th March, 01:00
  `https://codingcompetitions.withgoogle.com/codejam`
- **Event 3: Google Code Jam Round 1A**: 10th April, 02:00–04:30
- **Event 4: Google Code Jam Round 1B**: 25th April, 17:00–19:30
- **Event 5: Google Code Jam Round 1C**: 1st May, 10:00–12:30
- **Event 6: Google Code Jam Round 2**: 15th May, 15:00–17:30

Those interested in the competitive track must **register here** (you can join anytime):

`https://docs.google.com/spreadsheets/d/`

`1KMZx58SoE08g-l4usphtaLFnPhKzBDhTpa9PgixOok8/edit?usp=sharing`.

What will you gain?

- Improve your programming craft, and code beautifully.
- Learn the basics of Java.
- Learn the basics of object-oriented programming.
- Feel more confident in the code you write.
- Develop your first 500-1K LOC programs.

Your coder toolbox

As a future professional software programmer, you need a decent (virtual) equipment! Here a list of what you need (for this class):

- **Shell**: Linux-compatible bash shell (aka. console or terminal)
- **Editor**: Sublime Text (`https://www.sublimetext.com/3`)
- **Java compiler/runtime**: `java` and `javac` commands
  Get *Open Java Development Kit* (Open JDK) (`https://www.oracle.com/java/technologies/javase-downloads.html`)
- **Source code control**: Git with `git` command.
  Also Github (`https://github.com`) as a collaboration platform built on top of git.
- **Build automation tool**: Maven with `mvn` command.
- **Communication**: Discord app.

No IDE for now. You must use Sublime text. IDEs are quite complicated and you don't know what's going on. We'll use one later.

Getting started

Depending on your system, the ways to install the tools are a bit different. Please, follow these videos according to your operating system (password: Programm1ng):

- Linux (Ubuntu):
  `https://unilu.webex.com/unilu/ldr.php?RCID=63896a9159d2a523118c1f724251cd0f`
- Mac OSX:
  `https://unilu.webex.com/unilu/ldr.php?RCID=8caf3ec8a59b40fd5721e74142c27e5c`
- Windows:
  `https://unilu.webex.com/unilu/ldr.php?RCID=5e77758fb1d61a90dca84802062d5fd0`

Try out as soon as possible Exercise 1 of Lab 1.
You **CANNOT** stay stuck at this stage.
Ask on Discord for any problem.

Getting help

Google to search for information (*e.g.*, *Java docs*, *Stackoverflow*, . . .).

Discord (`https://discord.gg/SqarkmNQHe`) will be the privileged communication tool for questions.
Answer the questions of your peers, Mathi and I will answer too.
Here the different channels:

- `#tools`: for any installation trouble, *e.g.*, you can't run a Java program, and questions relevant to tooling.
- `#code`: all questions relevant to the code (from labs or classes).
- `#competition`: for the competitive track (UVa problems) and events (Google Hash Code, Code Jam).

By mail if your question is personal: `pierre.talbot@uni.lu`.

Do everything you can to find answers to your questions.

Resources

- **The Small Programming Handbook**: Cheat sheets on git, shell, Java pitfalls, Java conventions,... *Updated regularly* on `https://www.overleaf.com/read/tqxpqfwbbccc`
- Slides and recorded lectures, live coding and code analysis sessions.
- Tutorial on various topics inside the labs.

References

General Programming

- **Clean Code: A Handbook of Agile Software Craftsmanship**, Robert C. Martin
- Agile Software Development, Principles, Patterns, and Practices, Robert C. Martin
- Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
- The Mythical Man-Month: Essays on Software Engineering, Frederick Brooks

Java

- Effective Java 3rd Edition, Joshua Bloch
- Core Java Volume I - Fundamentals, Eleventh Edition, Cay S. Horstmann