Programming Languages 2 - 2023

Louis Botha, `louis.botha@tuni.fi`

Why is Coffee Called Java? (A Historical Relic)

:zoom: ZOOM | ::slack: SLACK | :github: GITHUB | :github-class: CLASSROOM | :dropbox: DROPBOX | :google_sheets: SHEET

Attendance & Exercises Sheets
:google_sheets: 22I260EA SHEET | :google_sheets: 22I260EB SHEET

> *:attention: Permissions will be granted in the first lecture*

Slack

Slack is basically a messaging app on steroids and is one of the standard communication tools used in companies, so it will also be the main communication channel for the course.

Slack workspace invitation link.

Each group have a private ::slack: channel:
::slack: 22I260EA
::slack: 22I260EB

The ::slack: general channel will be used for announcements and information.

Note: Custom emojis are allowed 😉

🎯 Goals

- After completing the course, the student will be able to write Java applications using object-orientated programming.

✅ Outcomes

- Student understands and can explain the concepts of object-oriented programming.
- Student is able to implement Java programs and applications by utilising object-oriented design.
- Student can develop programs with the Java programming language.

📅 Agenda and Deadlines

All lessons will be on campus.

Week	Topic	Lessons & Deadlines
02	**Kick Off &** **Introduction to Tools, CLI and Java**	Group EA & EB Mon 12:15 - 14:00

	💻 Exercises: • 1 - Introducing Slack ::slack: • 2 - Installing the development tools ⚒️ • 3 - Introducing the CLI :terminal: • 4 - Setting up Git ::git: and GitHub :github: • 5 - Getting started with GitHub :github: • 6 - Hello Java :java:	Group EA Tue 14:15 - 16:00 Group EB Fri 11:15 - 13:00 Deadline Mon 16.1 @ 12:00
03	**Variables, Conditional Logic, For/While, User Input** 📜 Material • Week 03 :github-class: Classroom • Assignment • Solutions	Group EA & EB Mon 12:15 - 14:00 Group EA Tue 14:15 - 16:00 Group EB Fri 11:15 - 13:00 Deadline Mon 23.1 @ 12:00
04	**OOP - Classes, Objects, Constructors & Encapsulation** 📜 Material • Lesson 1 :github-class: Classroom • Assignment • Solutions	Group EA & EB Mon 12:15 - 14:00 Group EA Tue 14:15 - 16:00 Group EB Fri 11:15 - 13:00 Deadline Mon 30.1 @ 12:00
05	**OOP - Subclasses & Inheritance**	Group EA & EB

	❓ Take a Guess 📜 Material • Lesson 1 • Workshop :github-class: Classroom • Assignment • Solutions	Mon 12:15 - 14:00 Group EA Tue 14:15 - 16:00 Group EB Fri 11:15 - 13:00 Deadline Mon 6.2 @ 12:00
06	**OOP - Abstract Classes, Interfaces & Polymorphism** 📜 Material • Lesson 1 • Clean Code Activity :github-class: Classroom • Assignment • Solutions	Group EA & EB Mon 12:15 - 14:00 Group EA Tue 14:15 - 16:00 Group EB Fri 11:15 - 13:00 Deadline Mon 13.2 @ 12:00
07	**OOP - Inner, Anonymous Classes & Lambda Expressions** ❓ Take a Guess 📜 Material • Lesson 1 :github-class: Classroom • Assignment • Solutions 🚥 Retrospective	Group EA & EB Mon 12:15 - 14:00 Group EA Tue 14:15 - 16:00 Group EB Fri 11:15 - 13:00 Deadline Mon 20.2 @ 12:00

	• Mural Board	
08	**Collections** • Arrays • ArrayList • HashMap • HashSet 📜 Material • 1d and 2d Arrays • Collections • Workshop & Solutions :github-class: Classroom • Assignment • Solutions	Group EA & EB Mon 12:15 - 14:00 Group EA Tue 14:15 - 16:00 Group EB Fri 11:15 - 13:00 Deadline Fri 17.3 @ 8:00
09	🌴 **SPRING BREAK** 🌴	
10	**Collections &Java Refresh** 🌲 Green Coding Learning Café :github-class: Classroom • Java Refresh Assignment 🏆	Group EA & EB Mon 12:15 - 14:00 Group EA Tue 14:15 - 16:00 Group EB Fri 11:15 - 13:00 Deadline Fri 17.3 @ 8:00
11	**Strings & StringBuilder** **JavaDoc** **Error Handling**	Group EA & EB Mon 12:15 - 14:00 Group EA

	📜 Material • Java String & StringBuilder • Java Doc • Error Handling :github-class: Classroom • Assignment • Solutions	Tue 14:15 - 16:00 Group EB Fri 11:15 - 13:00 Deadline Mon 20.3 @ 12:00
12	**Java Streams** 📜 Material • Java Streams • Java Optionals • Workshop :github-class: Classroom • Assignment • Solutions	Group EA & EB Mon 12:15 - 14:00 Group EA Tue 14:15 - 16:00 Group EB Fri 11:15 - 13:00 Deadline Mon 27.3 @ 12:00
13	**Java Practice** 📜 Material • Practice questions brief	Group EA & EB Mon 12:15 - 14:00 Group EA Tue 14:15 - 16:00 Group EB Fri 11:15 - 13:00 Deadline Mon 3.4 @ 12:00
14	**Software Testing & JUnit5** 📜 Material	Group EA & EB Mon 12:15 - 14:00

	• Software Testing • Final Exercise Brief :github-class: Classroom • Unit Testing • Final exercise Retrospective • Mural Board	Group EA & EB Tue 14:15 - 16:00 Easter Friday
15	**Final exercise** • Support in Slack/Teams/Zoom • Support on campus on agreement	Easter Monday Group EA Tue 14:15 - 16:00 Group EB Fri 11:15 - 13:00
16	**Exam** Exam starts **12:00** sharp. • If you are late, you will **NOT** be let in **Final exercise**	Mon 17.4. @ 12:00 - 14:00
17	**Exam - Retake 1** **Final exercise** Final Exercise Deadline	Mon 24.4. @ 12:00 - 14:00 Sun 30.4 @ 23:59
18	**Exam - Retake 2** **Final Exercise Deadline**	Tue 2.5 @ 14:00 - 16:00 **Tue 2.5.2023 @ 8:00**

> *:attention: Note, this is a plan and plans might change.*

👷 Groups

The groups will be according to the arrival class designation.

Group 22I260EA
Monday 12:00 - 14:00
Tuesday 14:00 - 16:00

:google_sheets: 22I260EA SHEET | ::slack: 22I260EA

Group 22I260EB
Monday 12:00 - 14:00
Friday 11:00 - 13:00

:google_sheets: 22I260EB SHEET | ::slack: 22I260EB

> *:attention: If you are not in one of these admission groups by default, you can decide on which group fits your course schedule best and inform me.*

💻 Programming Tools

You will need to have installed Java 11 or newer. Will be installed as part of an exercise.

Also some good code editor, like Visual Studio Code is needed. I will be using Visual Studio Code. Will be installed as part of an exercise.

Be sure that `java` and `javac` commands work. Will be installed as part of an exercise using a JDK manager.

TO DECIDE: More advanced tools, like **IntelliJ IDEA**, **Eclipse** or **Netbeans** are **NOT** needed and used in this course.

✅ Course Requirements

- Must **PASS** the **exam** (min 40%)
- Do **assignments/exercises**
  - Returned before next lesson to GitHub classroom
  - The final exercise will be a larger program counting more points

> *:attention: Theoretically it is possible to pass the course by only doing the exam. The implication is that your maximum grade for the course will be 3.*

Exam Grading

Exam contains programming assignments. Exam grade is calculated:

```
[ 0%, 40%[  =>  0p
[40%, 60%[  =>  1p
[60%, 80%[  =>  2p
[80%, 100%] =>  3p
```

Exam will be **"Open Book"**, so you can bring any paper material (book or printed) with you.

> *:attention:*
> *-**Internet usage is prohibited (not allowed) during the exam.***
> *-Not allowed to have GitHub or Dropbox paper open. You will need to make some notes for yourself from the exercises and handouts for some of the main concepts.*
> *-Not allowed to have any other applications than a terminal and code editor open*

Exercise Grading

Course will include weekly exercises. Exercises will most of the time be started during the lessons but those that are not complete are homework. Completed exercise will be marked during lessons on a weekly basis. Exercise completion will be converted to a percentage and the percentage converted to credits:

```
[ 0%, 30%[  => 0p
[30%, 70%[  => 1p
[70%, 100%[ => 2p
```

Exercise Quizzes

There will be random multiple choice quizzes related to the subject of the week and exercises. If you have done the exercises yourself, you should pass. Quiz marks will be added to the :google_sheets: exercise sheet and affect the exercise grade.

Overall Grade

Student gets from 0 to 2 points for exercise activity and 0 - 3 points from the exam. Summing these both together will be the course grade.

💾 Plagiarism

Students are allowed to discuss the exercises and projects in general and you are allowed to help and advise.

Copying of any line of code will be considered as plagiarism.

All submitted exercise and projects will be checked with a plagiarism tool and analysed for similarities.

> *:attention: The tools are quite efficient.*
> *Just renaming a few variables and methods will not prevent being caught out.*

Apart from the tools being used, the evaluator will also check for obvious similarities.

Plagiarized exercises and projects will be rejected.

TAMK rules related to fraud.

Improvement Ideas & Suggestions

- Better exercise management
  - Use GitHub Classroom + Moodle Assignments
- Way to pass course based on previous programming knowledge
- First 2 or 3 lessons should be about Java Basics still
  - Don’t rely too much on the Programming 1 course
- Split Collections into own lessons
- More presentations, also make presentation about code and implementations
- Flip some lessons, a video lesson before hand to set the stage for the lesson.
- Move course to Moodle