Introduction and course organization

https://github.com/heig-vd-dai-course

Web · PDF

L. Delafontaine and H. Louis, with the help of GitHub Copilot.

Welcome to the Développement d'applications internet (DAI) course!

Who are we?

Ludovic Delafontaine

Both classes

Mail · GitHub

Hadrien Louis

Monday mornings

Mail · GitHub

Géraud Silvestri

Friday mornings

Mail · GitHub

What to expect to learn from this course

What does GAPS say

- Fiche d'unité
- Descriptif de module
- Programme de formation
- 3 ECTS credits = ~75-90 hours
    - 48 hours in class (= 64 periods of 45 minutes)
    - ~27-42 hours outside of the class - We try our best to ensure you do not have to work that much outside of the class

> ### *Connaissances préalables recommandées*
>
> *L'étudiant-e doit connaître et savoir utiliser les notions suivantes :*
>
> - *Programmation (Java, C) ;*
> - *Programmation orientée objet ;*
> - *TCP/IP.*
>
> *Les unités d'enseignement PRG2, RXI et POO permettent d'acquerir ces connaissances.*

> ### *Objectifs*
>
> *A l'issue de cette unité d'enseignement, l'étudiant-e sera capable de :*
>
> ***Programmation réseau***
>
> - *être capable de concevoir une application client-serveur ;*
> - *être capable d'implémenter un client et un serveur en utilisant l'API Socket dans différents langages ;*

> “ ***Protocole HTTP***
>
> - *connaître les concepts principaux du protocole ;*
> - *être capable de concevoir et réaliser une infrastructure HTTP avec un reverse proxy et plusieurs serveurs ;*
> - *être capable d'implémenter le protocole en utilisant l'API Socket ;* ”

> ### *Protocoles de messagerie*
>
> - *connaître les principaux protocoles relatifs à la messagerie électronique ;*
> - *être capable d'implémenter un client de messagerie simple ;*

> ### *Protocoles de transfert de fichiers et d'accès à distance*
>
> - *connaître les protocoles de transfert de fichiers et d'accès à distance, ainsi que leurs principales utilisations (y compris tunneling/forwarding).*

Topics

You will learn the following topics during this course:

- Network programming (inputs/outputs, encodings, TCP and UDP)
- Application-level protocols (SMTP, SSH, HTTP and your own)
- Web infrastructures (reverse proxy, load balancer, sticky sessions)
- How to read and write technical documentation

At the end of the course, you will be able to create applications that can communicate over the network!

Technologies

You will learn the following technologies during this course:

- Git and GitHub
- Markdown
- Java for network programming
- Docker & Docker Compose
- Network utilities
- The terminal ❤️

Why do you have this course

This course defines the basics of network communication and how all these communications are programmed.

At the end of the course, you will know how to define, program and deploy network applications, how to interact with them, and the different elements to pay attention to make robust applications.

Whether you are in software, security, data science, embedded or network, you will have to deal with network applications (APIs, devices, etc.). This course will give you a solid grounding in this world.

Course planning - Calendrier académique

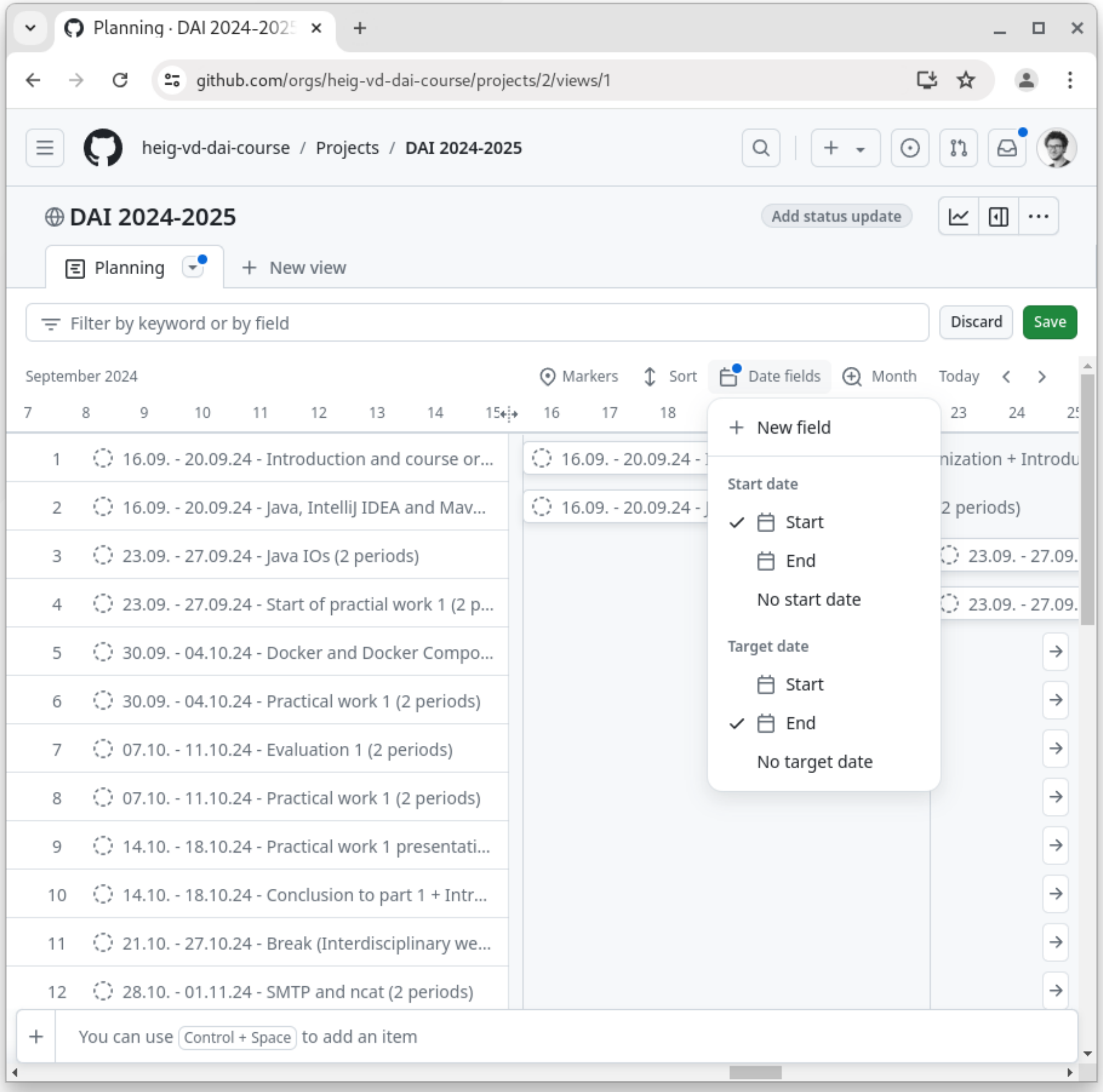
64 periods are planned for this course, divided in 3 main parts:

- Part 1 - Input/output processing (18 periods)
- Part 2 - Network programming with TCP and UDP (24 periods)
- Part 3 - Network programming with HTTP (20 periods)

The entire course planning for the semester is available at https://github.com/orgs/heig-vd-dai-course/projects.

The planning is subject to change. We will do our best to inform you in advance if there are any changes.

*Set the **Start date** and the **End date** fields as shown to display the course planning correctly!*



Start date
- ✓ Start
- End
- No start date

Target date
- Start
- ✓ End
- No target date

1 16.09. - 20.09.24 - Introduction and course or...
2 16.09. - 20.09.24 - Java, IntelliJ IDEA and Mav...
3 23.09. - 27.09.24 - Java IOs (2 periods)
4 23.09. - 27.09.24 - Start of practial work 1 (2 p...
5 30.09. - 04.10.24 - Docker and Docker Compo...
6 30.09. - 04.10.24 - Practical work 1 (2 periods)
7 07.10. - 11.10.24 - Evaluation 1 (2 periods)
8 07.10. - 11.10.24 - Practical work 1 (2 periods)
9 14.10. - 18.10.24 - Practical work 1 presentati...
10 14.10. - 18.10.24 - Conclusion to part 1 + Intr...
11 21.10. - 27.10.24 - Break (Interdisciplinary we...
12 28.10. - 01.11.24 - SMTP and ncat (2 periods)

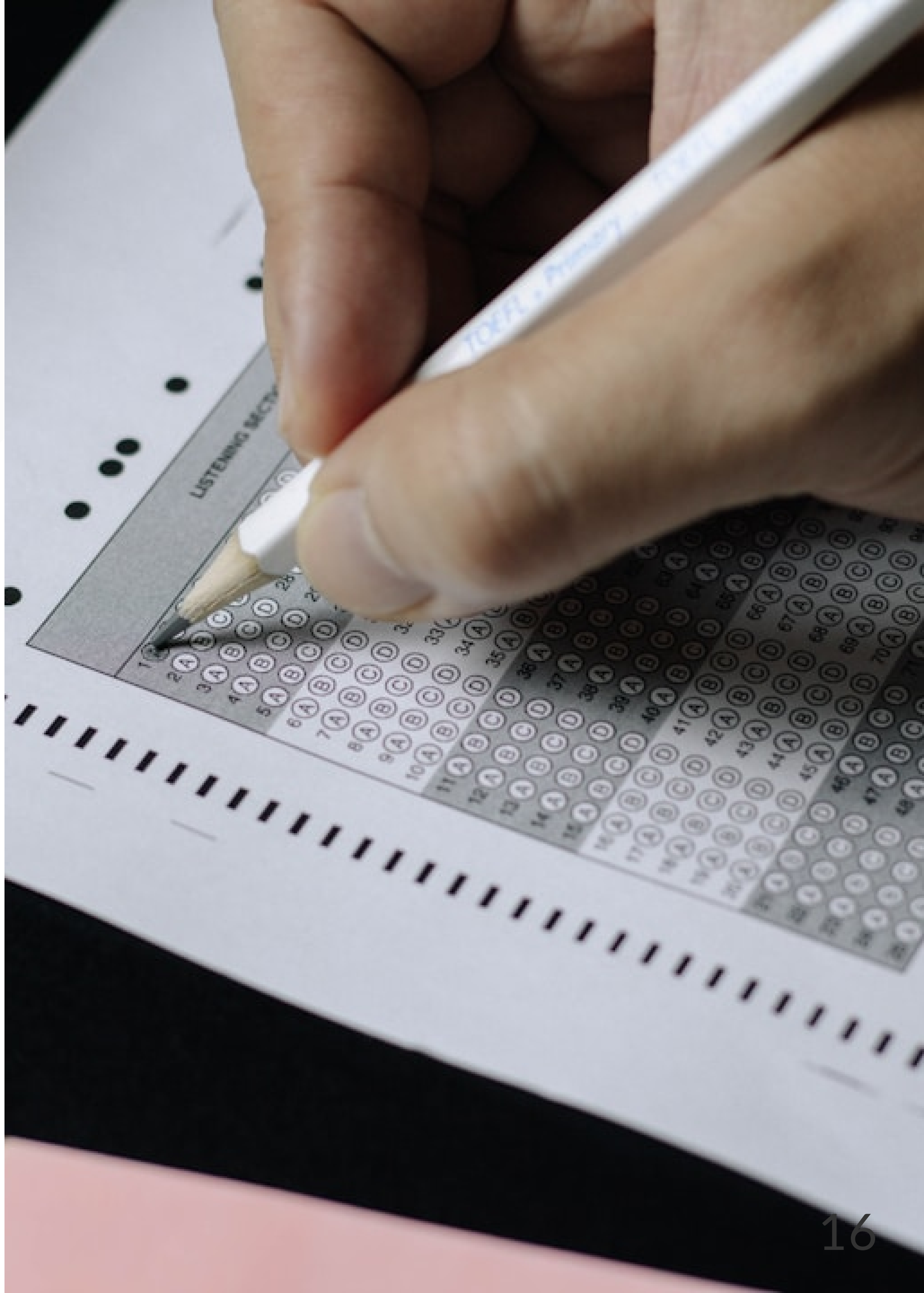
You can use Control + Space to add an item

Course sequence

1. Overview of the subject meant to be short and concise (us)
   - Sessions of 20-30 minutes - Dynamic and interactive
2. Practice the content with theoretical and practical parts (you)
   - Prepare you for the practical work - We are available to help
3. Practical work (you)
   - Freedom of the subject - The course materials help you
4. Feedback and discussions (you - very important to us)
   - You will be able to give feedback and help us improve the course

Evaluation

- 3 practical works (20%) - done in groups
- 3 evaluations (30%) - done individually
- 1 exam (50%) - done individually



You can change groups between practical works.

We will try our best to give you your grades and feedback as quickly as possible so you can improve for the next work.

We can be flexible on the deadlines if you have a good reason. Please let us know as soon as possible if you have any difficulties.

Communication channels

Feel free to use any of these communication channels for any kind of questions related to the course:

- GitHub Discussions (you will experiment these very soon)
- In person
- Email
    - ludovic.delafontaine@heig-vd.ch
    - hadrien.louis@heig-vd.ch
    - geraud.silvestri@master.hes-so.ch

Where to find the course material

Everything will be on the GitHub organization created for this course: https://github.com/heig-vd-dai-course.

More specifically, the course material is available in the `heig-vd-dai-course` Git repository.

We will not use Microsoft Teams nor CyberLearn. We want to keep it simple and efficient for you, and for us. One place to find it all.

We accept issues and pull requests if you spot anything weird! 🙂

The course material is available in several formats:

- **Markdown format** (recommended). You can read it directly on GitHub or on your computer with your favorite text editor.
- **PDF format**. We are aware that the generated PDFs are not perfect, sorry about that.

These formats are generated from the same source files. There are no differences between them. Use the one you prefer!

Chapters marked as *"work in progress (WIP)"* in the main README file are still in review and might change.

Our wishes for this course

What we want this course to be

- A place to learn
- A place to experiment and where it is OK to make mistakes
- A place to share and discuss
- A place where you are able to do things and master them
- A place where you can express your opinion
- It is OK if you are not feeling well. We are here if you need to talk.

➡️ A place where you do not loose your time (efficient and useful)

What we don't want this course to be

- A place you don't want to come
- A place you don't feel comfortable/safe
- A place where you can't ask questions and/or do mistakes
- A place where you can't take anything away

➡️ If you have a problem with the course, we will do our best to fix it.

We will take presences during this course. If you are done, you are free to go. However, we would appreciate if you try to help your classmates instead!

Responsibilities and expectations

The teaching staff responsibilities

- You have acquired the necessary knowledge of this course
- You are prepared for the professional world
- You are able to work on your own, and in a team
- Your feedback is taken into account
- We are available to help you and answer your questions
- We try to give you accurate and useful feedback for you to improve

Your responsibilities

- Read and understand the course material
- Do the practical content given in the course material
- You can give feedback and suggestions
- You are responsible for your own learning:
  - If you have any questions, ask them
  - If you have any difficulties, let us know
  - If you have any suggestions, share them

Let's work together to give you the best course possible.

"Un cours en anglais..?"

All the written material will be in English. The teaching staff will speak in French. The evaluations and exam will be in French.

Our field is mostly made in English so this is a great opportunity to practice.

If you have any difficulties with English, please let us know.

"What about ChatGPT, GitHub Copilot, etc.?"

You can use all the tools you want (personal notes, Internet, AI tools, etc.), **except during evaluations and the exam**. You will be able to use these in the real world, so better learn how to use them now. However, they are here to help you, not to do the work for you:

- You **must be able to explain what you did and why you did it**
- You **must state when you used external help (tools and sources)**
- You **must be able to do the work on your own.** These tools come and go ($$$, end of life, etc.), you should not entirely rely on them.

"Linux, macOS or Windows?"

The HEIG-VD recommends Windows for your studies. This is mostly because other departments heavily rely on Windows-only products.

From our experience, software development on Windows can be more difficult than Linux or macOS.

If you are on Windows, follow the guide we created to set up a development environment with Windows Subsystem for Linux (WSL): *[Set up a Windows development environment]*.

This guide will help you to set up a Linux environment on your Windows computer to be able to follow the course (and beyond).

If you are on macOS, we recommend you to install Homebrew and use it to install the tools we will use during this course.

You can install Homebrew by following the instructions on the Homebrew website.

You can always install Linux on your computer if you want to. Ubuntu, Debian, or Fedora are recommended.

Whatever operating system you are using, **you should read the** ***Considerations for a development environment*** **guide**.

This guide will help you to set up your development environment in a way that will make you more productive and efficient.

"I have an ARM computer, is this an issue?"

If you have an ARM computer (Apple Silicon M1/M2/M3 for example), some parts of the course might be a bit more tricky. We will try our best to test all our course material but we cannot guarantee that everything will work as expected on the first try.

We might come to you to ask you to test some things for us before the next course that requires it.

If you encounter any issue, please let us know as soon as possible.

Questions

Do you have any questions?

Sources

- Main illustration by NASA on Unsplash
- Illustration by Kenny Eliason on Unsplash
- Illustration by Nguyen Dang Hoang Nhu on Unsplash
- Illustration by Fer Nando on Unsplash
- Illustration by Brett Jordan on Unsplash