

## The Rust programming language Summer 2024 / 2025

### Exercises

Number 3, Discussion: 2025 May 05

## 1 IO

1. Let's build a program that reads a file and filters out comments.

- Write a program that takes two `Strings` as a parameter (let's name them "filename" and "comment\_start"). Use `clap` (`cargo add clap --features derive` to install the `clap` dependency and the ability to use `derive`).
- Open the file "filename". Read it line by line. If a line does not start with "comment\_start", print it.
- Once the file is processed, write a report in a JSON file named "report.json". The report should contain the name of the file that was processed, the total number of lines, and the number of comment lines.

Hint: add this in your cargo.toml

```
serde = { version = "1.0", features = ["derive"] }  
serde_json = "1.0"
```

- New requirement just dropped: the user wants to decide the name of the report file. Add a "report" parameter to your program that lets them do exactly that.
- Document the parameters of your program. Verify that the `--help` parameter added by `clap` is informative enough.

## 2 Traits and generics

1. You are in charge of writing a software that keeps track of a warehouse's stock.

- Write a `Product` trait. Each product has different properties that you can read:
  - a `f32` price (given by `product.price()`)
  - a `u32` quantity (`product.quantity()`)
  - a `u8` "appreciation score" given by the consumers (`product.score()`)
  - a `String` description (`product.description()`)
- The shop owner wants to know the value of any `Product` he has in stock (just quantity  $\times$  price). Write a method `value()` with a default implementation.
- Implement the trait for 3 structs: `Food`, `Furniture`, and `Cutlery`.
- Write a function `print_product` that takes any `Product` and prints a small report about it (description and value left).