Pseudocode

How do you solve a complex problem?

Two approaches

Bottom-up: First build concrete small pieces, then put them together into higher-level pieces

Top-down: Start with high-level abstract pieces, and decompose into more concrete pieces

Bottom-up Programming

How to solve a problem:

(0) Decompose the problem into smaller more easily manageable pieces.

(1) First, write small, easily testable pieces of code (often functions)

(2) Write slightly more complicated pieces of code that call those small pieces

(3) Repeat until you've built up to the highest level!

Top-down Programming

How to solve a problem:

(0) Decompose the problem into smaller more easily manageable pieces.

(1) First, write the final function that you need.

(2) In that function, call other helper functions that you haven't created yet.

(3) Fill out those helper functions by having them call other helper functions.

(4) Repeat until functions don't need helper functions anymore, and you're done!

Bottom-up Scorekeeper

Bits and pieces I need:

- score printer
- score file reader
- score file writer
- score updater

First, write each of those bits individually, then add some top-level function to call each of those.

Top-down Scorekeeper

Two important top-down tools:

Pseudocode: write out code in English

Stub functions: functions that compile, but just return a dummy value

Pseudocode

```
load file function:
    open file
    for each line in the file
        if that line is a string
            create a new key in the dictionary
        if that line is an integer
            add a score to the last player
```

Stub functions

```
def load_score_file():
    # TODO Actually load a file
    return { "enne": [20, 40] }

def save_score_file(d):
    pass # TODO Actually save

d = load_score_file()
print_scores(d)
update_scores(d)
print_scores(d)
save_score_file(d)
```