

Designing codes

Programming Concepts in Scientific Computing
EPFL, Master class

November 15, 2023

#1 Scientific question

#2 Problem formulation

#2 Problem formulation

- ▶ Mathematics
- ▶ Identify inputs/outputs

#3 Algorithms description

#3 Algorithms description

- ▶ Decompose program in sub-parts

#3 Algorithms description

- ▶ Decompose program in sub-parts
- ▶ Choose algorithm to solve the sub-parts

#3 Algorithms description

- ▶ Decompose program in sub-parts
- ▶ Choose algorithm to solve the sub-parts
- ▶ Choose data structures

#3 Algorithms description

- ▶ Decompose program in sub-parts
- ▶ Choose algorithm to solve the sub-parts
- ▶ Choose data structures
- ▶ Identify polymorphic code: class diagram

#4 Implementation

#4 Implementation

- ▶ Decide where the code is hosted (for backups and revisions)

#4 Implementation

- ▶ Decide where the code is hosted (for backups and revisions)
- ▶ Decide a coding convention (question of style)

#4 Implementation

- ▶ Decide where the code is hosted (for backups and revisions)
- ▶ Decide a coding convention (question of style)

example:

<https://google.github.io/styleguide/cppguide.html>

#4 Implementation

- ▶ Decide where the code is hosted (for backups and revisions)
- ▶ Decide a coding convention (question of style)
example:
<https://google.github.io/styleguide/cppguide.html>
- ▶ Identify existing software for any sub-part

#4 Implementation

- ▶ Decide where the code is hosted (for backups and revisions)
- ▶ Decide a coding convention (question of style)
example:
<https://google.github.io/styleguide/cppguide.html>
- ▶ Identify existing software for any sub-part
- ▶ Decide a source documentation format

#4 Implementation

- ▶ Decide where the code is hosted (for backups and revisions)
- ▶ Decide a coding convention (question of style)
example:
<https://google.github.io/styleguide/cppguide.html>
- ▶ Identify existing software for any sub-part
- ▶ Decide a source documentation format
- ▶ Program the thing

#4 Implementation

- ▶ Decide where the code is hosted (for backups and revisions)
- ▶ Decide a coding convention (question of style)
example:
<https://google.github.io/styleguide/cppguide.html>
- ▶ Identify existing software for any sub-part
- ▶ Decide a source documentation format
- ▶ Program the thing
- ▶ Tests

#1 Scientific question

Many meteo devices measure constantly the temperature in Switzerland.

We wish to know the evolution of the average temperature in Switzerland, or the average temperature over a year for a given site, or some other combination of measure.

#2 Problem formulation

- ▶ Mathematics:

$$\bar{t} = \sum_i t_i(t) \cdot \Delta V_i$$

$$\bar{t} = \sum_t \sum_i t_i(t) \cdot \Delta V_i \Delta t$$

- ▶ Input: $t_i(t)$, output: \bar{t}

#3 Algorithms description

#3 Algorithms description

- ▶ Decompose program in sub-parts

#3 Algorithms description

- ▶ Decompose program in sub-parts

main \longrightarrow *Task 1* \longrightarrow *Task 2* $\longrightarrow \dots \longrightarrow$ *exit*

#3 Algorithms description

- ▶ Decompose program in sub-parts

#3 Algorithms description

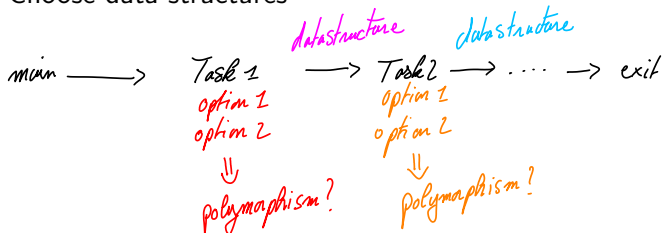
- ▶ Decompose program in sub-parts
- ▶ Choose algorithm to solve the sub-parts

#3 Algorithms description

- ▶ Decompose program in sub-parts
- ▶ Choose algorithm to solve the sub-parts
- ▶ Choose data structures

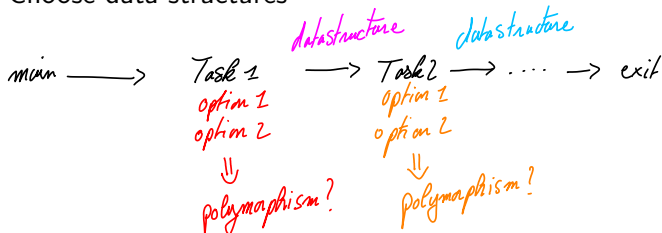
#3 Algorithms description

- ▶ Decompose program in sub-parts
- ▶ Choose algorithm to solve the sub-parts
- ▶ Choose data structures



#3 Algorithms description

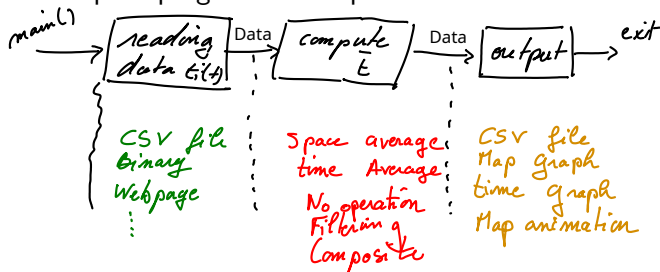
- ▶ Decompose program in sub-parts
- ▶ Choose algorithm to solve the sub-parts
- ▶ Choose data structures



- ▶ Identify polymorphic code: class diagram

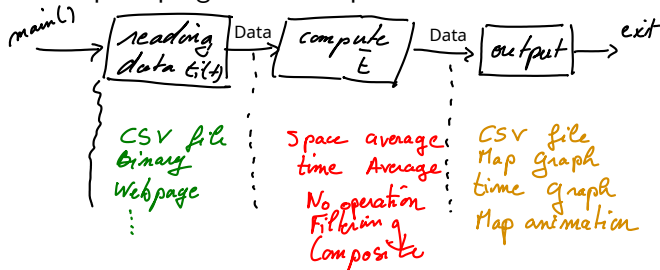
#3 Algorithms description

- ▶ Decompose program in sub-parts



#3 Algorithms description

- ▶ Decompose program in sub-parts



- ▶ Identify polymorphic code: class diagram for (Data, Readers, Compute, Output)
- ▶ Use the tool:

<https://codeskeleton-nyvvxr7lzakw9hvk3r5xya.streamlit.app/>

Take away message

Criterion for the projects

- ▶ Program compile and work
- ▶ Code factorization (polymorphic)
- ▶ Code documented with a short README
- ▶ Code documented with doxygen
- ▶ Code has tests