

Programming: The Next Step



Us

Alexander Savi

o.a.savi@uva.nl

Jasper Wijnen

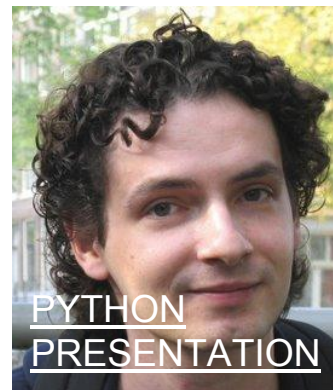
j.g.wijnen@uva.nl

Raoul Grasman

r.p.p.p.grasman@uva.nl

Han van der Maas

h.l.j.vandermaas@uva.nl



You

Research master psychology / behavioural data science / cognition in society

English / Dutch

Prior programming experience

Motivation

Course

Refresh and improve **your programming skills** while **independently** developing a **software program**, over the course of **four weeks**.

What to expect from course

Write a software program from start to finish

Document your software

Present your software and plans along the way

Workload of 21 hours per week

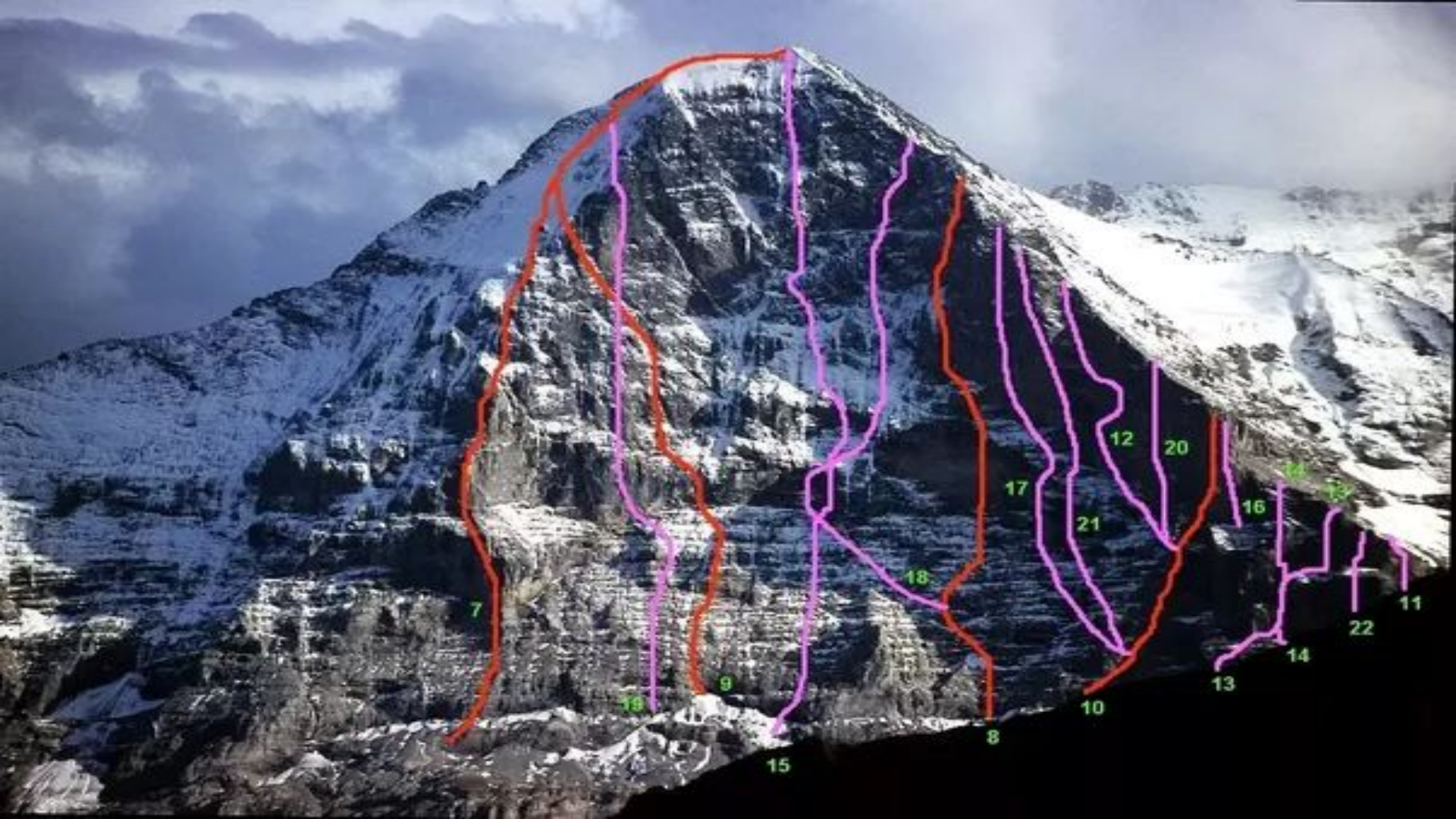
What to expect from us

2 hours of personal supervision

Programming superpower tips

No language instruction





7

19

9

15

18

8

10

17

21

12

20

13

14

16

22

11

Languages

[R](#) + [Shiny](#): statistical computing + interactive web applications

[Presentation](#) + [Mobile](#): experimentation

[MATLAB](#): numerical computing

[Python](#): general purpose

[JavaScript](#): web programming

[C](#): general purpose

[NetLogo](#): multi-agent modeling ([tutorial](#))

...

Topics: simulation & statistical software

[Stable marriage problem](#) / [college admission problem](#) ([Gale & Shapley, 1962](#))

Elo algorithm simulation ([Klinkenberg et al., 2011](#))

[Regression \(tree\) boosting](#) ([Drucker, 1997](#))

[Boltzmann machine](#) simulation

Suggested: R + Shiny, Python

[Pólya's urn model](#) simulation

Suggested: R + Shiny, Python, NetLogo

Network model of intelligence simulation ([van der Maas et al., 2017](#))

Suggested: NetLogo

Topics: experimentation software

[Visual search task](#) ...

[Simon task](#) ...

[Flanker task](#) ...

[Sternberg task](#) ...

[Attentional blink task](#) ...

[Balloon analogue risk task](#) ...

[Random dot motion task](#) ...

[Wisconsin card sorting task](#) ...

... with e.g., [task switching](#), adaptivity, gamification, AI opponent

Suggested: Presentation, Python

Topics

Propose your own topic

Make sure:

- you haven't worked on it before / it's not part of different graded project
- it can be done in 4 half-time weeks: make it scalable
- you'll create experimentation/simulation/statistical software

It's fine to reinvent the wheel, [to learn about the wheel](#)



Ideas

Schedule

	In class	At home
Tu 1	-	Language and topic selection, concept design
Tu 8	Present concept design (3m)	Implementation, update presentation
Tu 15	Present implementation (2m)	Implementation, update presentation
Tu 22	Present status and verification (2m)	Final report / manual
Th 31	Final presentation (5m)	-

Time 11:00-13:00; location G2.01

Slides on www.alexandersavi.nl/teaching/

Grading

60% software

- Functionality
- Coding style
- Within code documentation
- Consideration of superpowers (e.g., version control, testing procedure)

20% documentation

- Manual incl. task/technique description (requirements), flowchart of design, how-to for users

20% presentation

- Final 5-minute presentation
 - Preparation (e.g., working demo, within time limit)
 - Clarity (e.g., goal, design, implementation, functionality)
 - Quality of the software (based on what's shown)

NB. Your chosen topics will not be equally difficult, so effort will too be taken into account.

This Week: Concept Design



Concept: User Perspective

1. Narrative / description

- Describe in plain words who will be using it, how, and for what.

2. Graphical interface

- Draw an impression of the user interface.

3. Flowchart

- Create a flowchart of the options and actions of the user.

Concept: Software Perspective

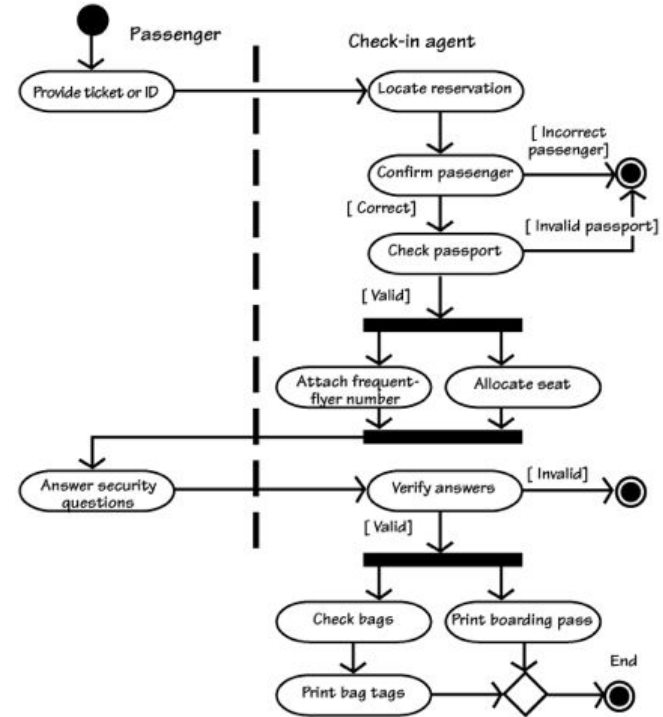
4. [Flowchart](#)

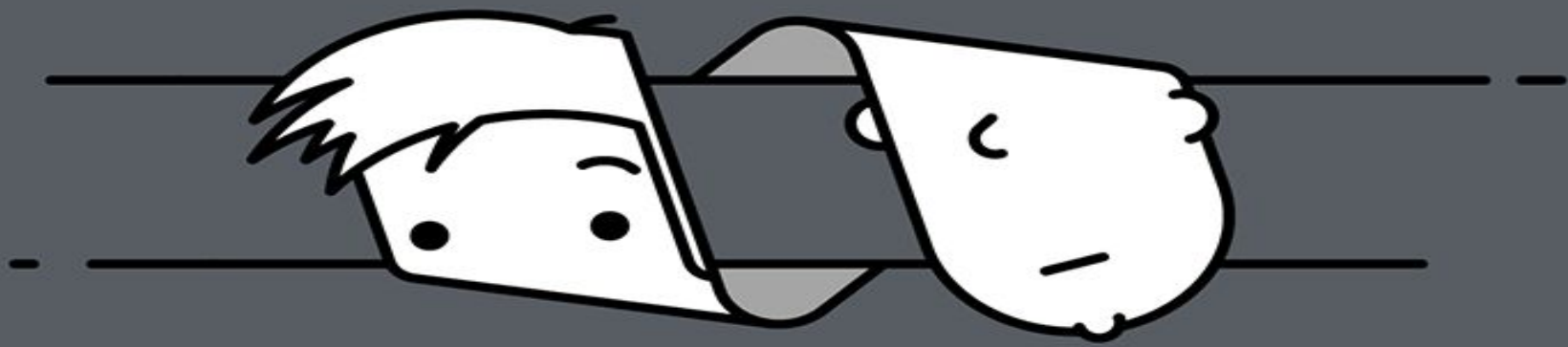
- Create a flowchart of the input, output, and internal functions of the program.

5. [Pseudocode](#) / [Structured English](#)

- Write the main functions in plain, structured language.

```
IF customer has a Bank Account THEN
  IF Customer has no dues from previous account THEN
    Allow loan facility
  ELSE
    IF Management Approval is obtained THEN
      Allow loan facility
    ELSE
      Reject
    ENDIF
  ENDIF
ELSE
  Reject
ENDIF
EXIT
```





"LET ME WRAP MY HEAD AROUND IT..."

Presentation

Create with [Google Slides](#)

Use 4 slides:

1. title page with your name, topic, language, supervisor
2. user perspective (minimal version)
3. software perspective (minimal version)
4. possible extensions / features

Share with o.a.savi@gmail.com

Present **next week** in **3 minutes**

Prepare: Version Control

Sign up for [GitHub](#)

Apply for [unlimited private repositories](#) (student email required)

Install [GitHub Desktop](#) (Linux users try [pre-release](#) version)

Already familiar with [Git](#) and using it? You can skip this!

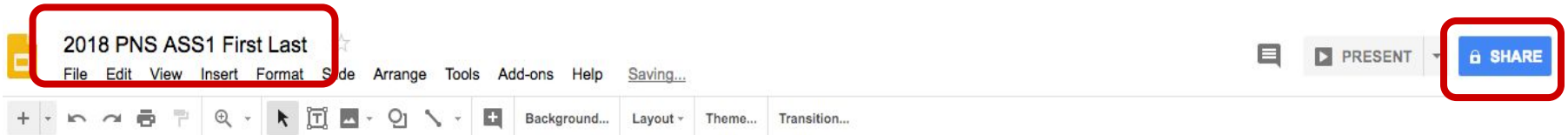
Deadlines

Choose topic and language: **Thursday, May 3, 17:00**

- <https://goo.gl/forms/V86SgOfX6KCdZgWH2>

Share presentation: **Tuesday, May 8, 9:00**

- Name it, share it



Be prepared for GitHub use: **Tuesday, May 8, 11:00**

Comments / feedback

We're here to **facilitate** your learning experience

Any **comments** or **feedback**?

Tell us **during** the course

You can do so **anonymously** on www.alexandersavi.nl/teaching/

Questions?



Happy designing!

