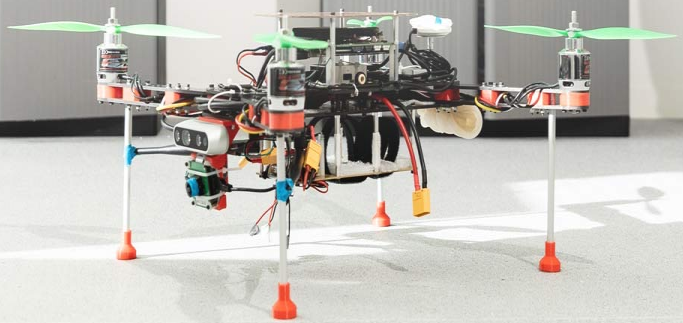
Procedural Programming

KEN1120
BCS1120



Enrique Hortal
Daniel Cámpora

Maastricht University | Department of Advanced Computing Sciences

Who is Enrique?

Research topics

- Machine Learning – Deep learning
- Affective computing
- (Bio)Signal processing

Teaching

- Computer Science / Programming
- Machine Learning (BSc and MSc)

I am Spanish…

- I have a strong Spanish accent
- It will not change (at least during this course)
- Sorry ☺

Enrique Hortal (he/him)
Assistant Professor
Department of Advanced Computing Sciences
enrique.hortal@maastrichtuniversity.nl
Paul-Henri Spaaklaan 1 (**room C4.020**)

Who is Daniel?

Research topics

- High Performance Computing (CPU / GPU)
- High Energy Physics reconstruction at CERN
- Applications of Machine Learning

Teaching

- Computer Science / Programming
- Software Engineering

Yo soy español

- Born in Sevilla, capital of the world cca. 1600
- I live in Maastricht with my girlfriend and piano
- I play chess, piano, board games and videogames
  I (used to) like hiking and snowboarding

Daniel Cámpora (he/him)
Assistant Professor
Department of Advanced Computing Sciences
d.camporaperez@maastrichtuniversity.nl
Paul-Henri Spaaklaan 1 (**room C4.032,** *I work mostly remote*)

Teaching Assistants (TAs)

- Spriha Joshi

- Annada de Freitas Sousa

- Abhimanyu Anand
- Adrien Bersia
- Ignacio Cadarso Quevedo
- Anna-Lena Krause
- Vasilis Papadakis
- Fivos Tzavellos

- Some 3rd year BSc students

Class Policies

- Class behavior
  - Tell us your preferred names/pronouns
  - We do not tolerate discrimination/violence

- Cheating/Honor Code
  - Plagiarism tools (also for the exams)
  - Violations will be reported when seated

Wooclap

How to participate?

1. Go to wooclap.com
2. Enter the event code in the top banner

Event code
PRPMODULE1

1. Send @PRPMODULE1 to 0970 1420 2908
2. You can participate

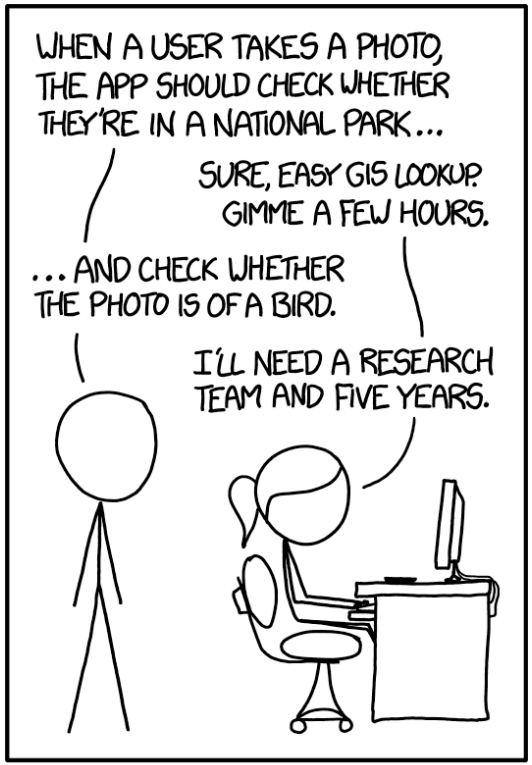
Course Goals

- To provide you with the introduction to computer science
  = Programming: how to get a computer to do stuff
  - Every computer science course after this one, will rely on this one!
    from artificial intelligence over mathematical simulation to bioinformatics and more …
- To show you how to translate one type of knowledge into a computer usable format
- To introduce you to Java as one possible programming tool

Why learn how to program?

"I don't want to spend my entire life programming"

"That's great, but you will need to know what is possible/easy/difficult and how it is done"



Algorithms: A running example

Recipe: making a chocolate soufflé:

as per Gordon Ramsey

Ingredients:

FOR THE DISHES
- 25g unsalted butter, for greasing
- finely grated chocolate

FOR THE CREME PATISSERIE
- 2 tbsp plain flour
- 2 tsp caster sugar
- ½ tsp cornflour
- 1 medium egg yolk
- 1 medium whole egg
- 4 tbsp milk
- 1 tbsp double cream
- 25g good-quality dark chocolate preferably 70% cocoa solids, broken in pieces
- 1 tbsp cocoa powder

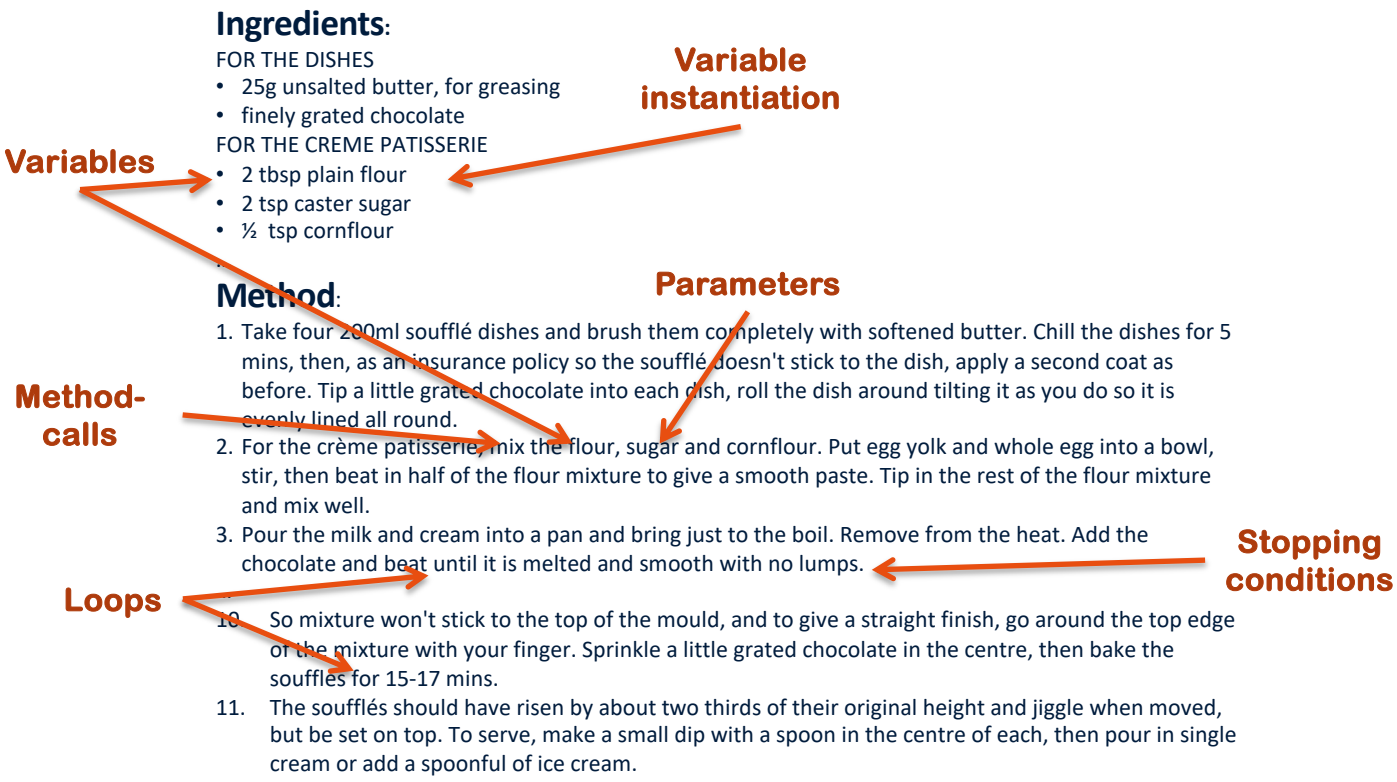
FOR THE EGG WHITES
- 6 medium egg whites
- 85g caster sugar
- single cream or ice cream, to serve

FOR THE GANACHE
- 4 tbsp double cream
- 50g good-quality dark chocolate preferably 70% cocoa solids, broken into pieces
- 1 tbsp cocoa

Method:

1. Take four 200ml soufflé dishes and brush them completely with softened butter. Chill the dishes for 5 mins, then, as an insurance policy so the soufflé doesn't stick to the dish, apply a second coat as before. Tip a little grated chocolate into each dish, roll the dish around tilting it as you do so it is evenly lined all round.
2. For the crème patisserie, mix the flour, sugar and cornflour. Put egg yolk and whole egg into a bowl, stir, then beat in half of the flour mixture to give a smooth paste. Tip in the rest of the flour mixture and mix well.
3. Pour the milk and cream into a pan and bring just to the boil. Remove from the heat. Add the chocolate and beat until it is melted and smooth with no lumps.
4. Gradually stir hot chocolate mix into paste. Return to pan. Cook, stirring, over a medium-low heat for 5 mins to a smooth, thick paste. Remove from the heat. Leave until cold, beating occasionally with a wire whisk.
5. Make the ganache: slowly warm the cream in a pan. Just before it boils, take off the heat and add chocolate. Beat constantly to a velvety texture, gradually sprinkling in the cocoa as it dissolves. Allow to cool.
6. Heat oven to 190C/fan 170C/gas 5. Whisk the egg whites to soft peaks with an electric whisk. Sprinkle in the sugar as you are mixing. Keep whisking to give stiff, firm peaks to give volume to the soufflés.
7. Mix crème patisserie and ganache in a large bowl. Stir in 2 tbsp of egg white. Carefully fold in a third of the rest, cutting through the mixture. Fold in another third (take care not to lose the volume); fold in the rest.
8. Spoon the mixture into the dishes to fill them by three-quarters, then gently press a spoon in to make sure it fills all the gaps. Fill the dishes to the top with the mixture, then bang each dish on to the surface so the mixture fills the sides.
9. Take a palette knife and pull it across the top of each dish so the mixture is completely flat. Take a little time to wipe any splashes off the outside of each dish, or they will burn on while cooking.
10. So mixture won't stick to the top of the mould, and to give a straight finish, go around the top edge of the mixture with your finger. Sprinkle a little grated chocolate in the centre, then bake the soufflés for 15-17 mins.
11. The soufflés should have risen by about two thirds of their original height and jiggle when moved, but be set on top. To serve, make a small dip with a spoon in the centre of each, then pour in single cream or add a spoonful of ice cream.

Relation to Programming?

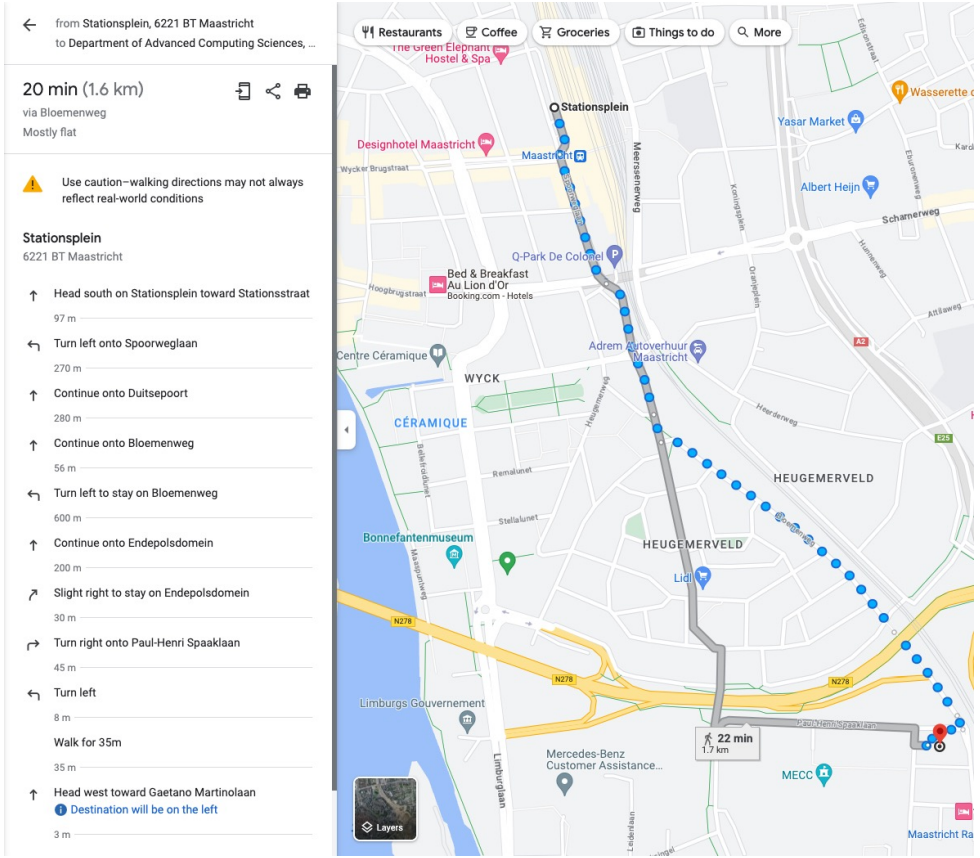


Algorithms

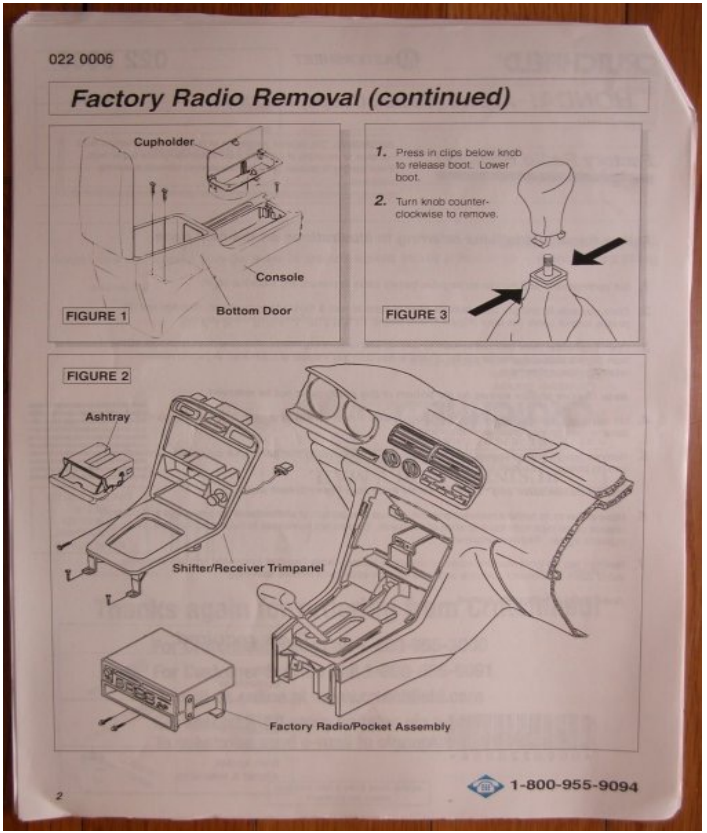
- "A step-by-step problem-solving procedure, especially an established, recursive computational procedure for solving a problem in a finite number of steps"
  - *From American Heritage® Dictionary of the English Language, 4th Edition*

- You've seen lots of algorithms before…

Example algorithms: Map directions



Example algorithms: Car radio removal



Incorrect algorithms

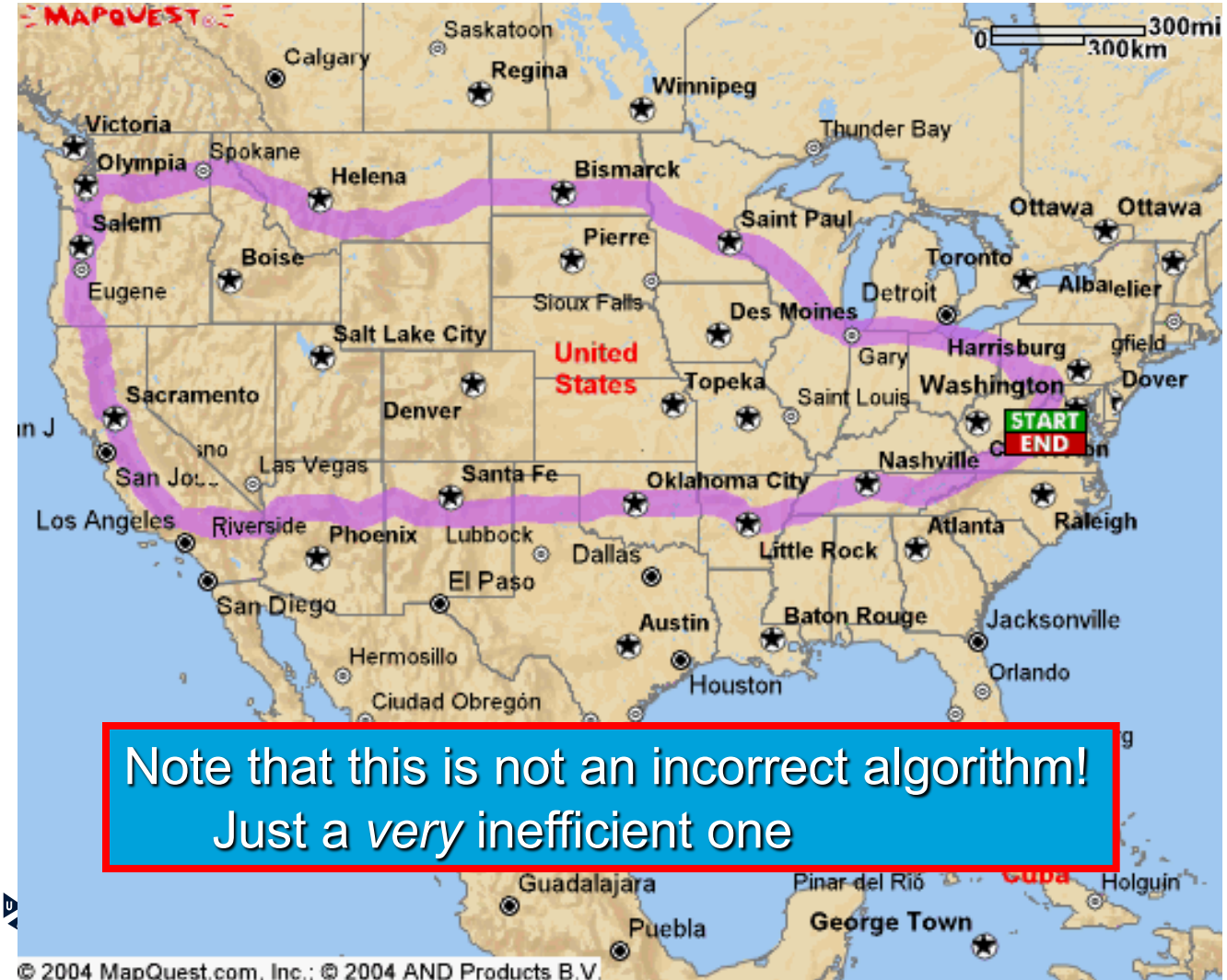
- Not all algorithms are “good”
- So then what makes an algorithm “bad”?
  - Can be wrong
  - Can be inefficient
  - Can never stop
  - Can have other problems as well…

Incorrect algorithms: Plain wrong

Incorrect algorithms: IKEA?

Inefficient algorithms: MapQuest directions

- Consider directions to get around grounds



Note that this is not an incorrect algorithm!
Just a *very* inefficient one

More recent examples

WORLD BRIEFING | AUSTRALIA

IPhone Leads Drivers to Middle of Nowhere in Australia

By The Associated Press

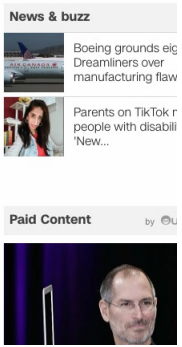
Dec. 12, 2012

Nearly 100 drivers followed a Google Maps detour -- and ended up stuck in an empty field

By Michelle Lou, CNN

Updated 1420 GMT (2220 HKT) June 27, 2019

Source: KMGH



Inefficient algorithms: Shampoo directions

level of conditioning for hair to move freely and fall naturally in place.

What You Get: Amazingly light and lively, clean hair that's manageable and moves naturally. Pert Plus has clean conditioning that leaves no unwanted residue – rinsing cleanly away without build-up. And because it's pH balanced, it's perfect for everyday use, even on permed or color-treated hair.

Go beyond clean and bring back lively hair.

Visit us at www.pertplus.com

DIRECTIONS: Gently massage into wet hair, lather, rinse, repeat

INGREDIENTS: Water, Ammonium Laureth Sulfate, Ammonium Lauryl Sulfate, Glycol Distearate, Sodium Chloride, Cocamide MEA, Dimethicone, Cetyl Alcohol, Fragrance, Sodium Citrate, Polymethacrylamidopropyl Trimonium Chloride, Sodium Benzoate, PEG 14M, Dihydrogenated Tallowamidoethyl Hydroxyethylmonium Methosulfate, Disodium EDTA, Citric Acid, Ammonium Xylenesulfonate, Methylchloroisothiazolinone, Methylisothiazolinone, Yellow 10, Blue 1.

Questions?
1-800-436-7378 TK

Made in USA of US and imported ingredients./Hecho en EE.UU. con ingredientes de EE.UU. e importados.
Distr. by/por
Procter & Gamble,
Cincinnati, OH 45202.
©2002
US Patent RE 34,584 ano 5,756,436.

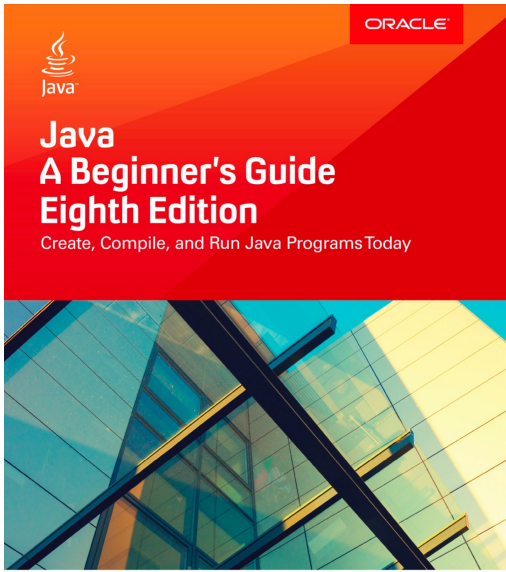
95051964

Course Material

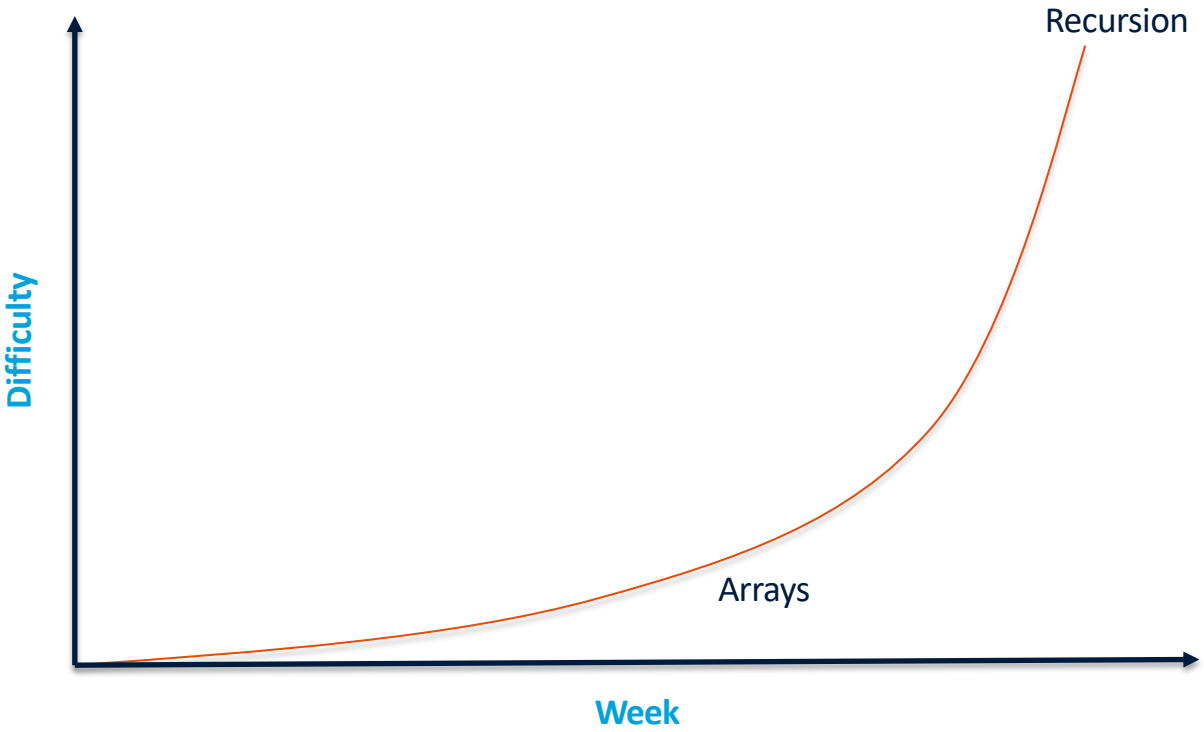
- Textbook
  Herbert Schildt, Java a Beginner's Guide (8th edition)

- Slides
- Quizzes + Homework

- Exercises/Practical Sessions



Course Content



Course Content

- Basic data types, variables
- Variables and methods
- Conditional execution
- Loops
- Arrays
- Recursion / Branching (For Project 1-1)

Schedule

Special sessions (Intro, recap, exam preparation, game award)
Live coding session
Office hours
Tutorial
Ask a TA
Binge! and Assignments (at your own)

Week	Day	Hours	Content	Type	Who	Book chapters
1	Tuesday 5/9	2	**Introduction to CS**	Lecture / Intro	Enrique	1,2,4,6 (check Canvas for details)
	at your own pace	1	**Variables and Methods**	Watch material	You!	
	Wednesday 6/9	2	>>	Live coding	Daniel	
	at your own pace	1	>>	Review material	You!	
	Thursday, 7/9	2	>>	Office hours	Enrique	
	Friday 8/9	1	>>	Tutorial	TA	
	at your own pace	6	>>	Assignment 1 (due 15/9)	You!	
2	at your own pace	1	**Conditionals**	Watch material	You!	2,3 (check Canvas for details)
	Tuesday 12/9	1	>>	Live coding	Enrique	
	at your own pace	1	>>	Review material	You!	
	Thursday, 14/9	2	>>	Office hours	Daniel	
	Friday 15/9	1	>>	Tutorial	TA	
	at your own pace	8	>>	Assignment 2 (due 22/9)	You!	
3	at your own pace	1	**Loops**	Watch material	You!	2,3 (check Canvas for details)
	Tuesday 19/9	2	>>	Live coding	Daniel	
	at your own pace	1	>>	Review material	You!	
	Thursday, 21/9	2	>>	Ask a TA	TAs	
	Thursday, 21/9	2	>>	Office hours	Daniel	
	Friday 22/9	1	>>	Tutorial	TA	
	at your own pace	8	>>	Assignment 3 (due 29/9)	You!	
4	at your own pace	1	**Arrays**	Watch material	You!	5 (check Canvas for details)
	Tuesday 26/9	2	>>	Live coding	Daniel	
	at your own pace	1	>>	Review material	You!	
	Wednesday, 27/9	2	**RECAP**	Q & A session	Daniel	
	Friday 29/9	1	>>	Tutorial	TA	
	Friday 29/9	2	>>	Office hours	Daniel	
	at your own pace	8	>>	Assignment 4 (due 6/10)	You!	
5	at your own pace	1	**Recursion**	Watch material	You!	6 (check Canvas for details)
	Monday 2/10	2	>>	Live coding	Enrique	
	Tuesday 3/10	2	**Branching**	Live coding	Enrique	
	at your own pace	1	>>	Review material	You!	
	Wednesday, 4/10	2	>>	Office hours	Enrique	
	Friday 6/10	1	>>	Tutorial	TA	
	Friday 6/10	2	>>	Ask a TA	TAs	
	at your own pace	10	>>	Assignment 5 (due 19/10)	You!	
6	Project 1.1					
7	Tuesday 18/10	2	**Revising all content**	Exam preparation	Daniel	
	at your own pace	10	Revising all content	Review material	You!	
	Wednesday 19/10	2	**Game award ceremony**	Special session	Enrique	
	at your own pace	10	Exam examples	Assignment 6 (due 19/10)	You!	
8	TBA	20	Study and Exam		You!	

Binge PrP at home!

- For each live coding lecture, we expect you to study (and/or watch) specific material that will prepare you for the actual lecture.
- It is important to study this material **before** the relevant live coding lecture!

- If you skip a live coding lecture, make sure to catch up with the content before showing up to the next practical session.

Live Coding/Lectures

- Lecture 1: Introduction to computer science
- Lecture 2: Variables and methods
- Lecture 3: Decisions (conditional execution)
- Lecture 4: Loops
- Lecture 5: Arrays
- Recap session
- Lecture 6: Recursion
- Lecture 7: Branching (For Project 1-1)
- PROJECT WEEK
- (not really a lecture): Exam preparation session
- (not really a lecture): …

Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7

Review material

- If you haven't watched the videos / study the material before the lecture, do it afterwards!

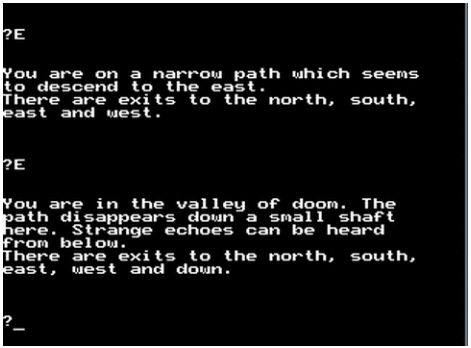
- Extra material and exercise to practice (see next slide)

Quizzes and Homework

- After each lecture, study the slides/book chapters and
  - try answering the **quizzes** (automatically graded!) and
  - attempt the **homework**
- Exercises in quizzes/homework include:
  - Explain concepts discussed in lectures
  - Find mistakes in programs
  - Run programs by hand
  - Design (small) solutions to (small) problems
  - …
- Goal: Prepare you for the (*real*) practical sessions
- Not graded, but important component

Tutorials

- Text-based adventure games

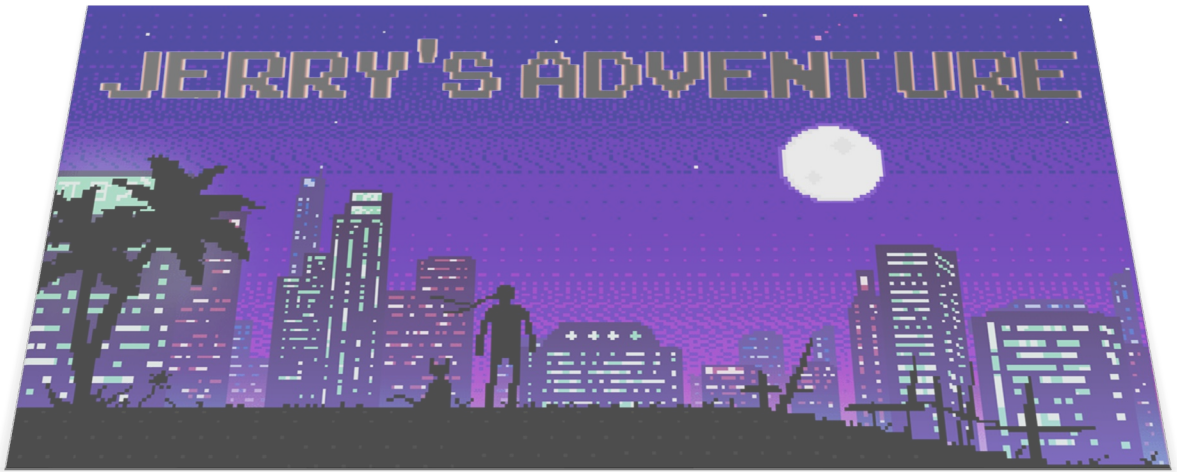


https://nonl.weebly.com/text-adventure-games.html

https://devpost.com/software/text-based-adventure-game-7a9x5g

Tutorials

- 5 tutorials/hands-on sessions (50-minute long)
- Small groups guided by a tutor (Teaching Assistant, aka TA)
- Development of a basic old-style fun adventure game, “Jerry’s adventure”



Tutorials

- This game is a good opportunity to practice before starting the assignments
- For you to train on, level adjusted to the learning curve of the classes
- Structure:
  - 5’ of silent reading of the assignment
  - 10’-15’ to “brainstorm” with the TA and start drafting a piece of pseudocode to solve the proposed assignment
  - 25’ to start translating the pseudocode into actual code, supported by the TA
  - 5’ to wrap up and establish future lines to improve the implementation of the game

Tutorials

- Timeline:
  - Session 1: 8:30 – 9:20
  - Tag team (+ TA break)
  - Session 2: 9:40 – 10:30
  - Tag team (+ TA break)
  - Session 3: 11:00 – 11:50
  - Tag team (+ TA break)
  - Session 4: 12:10 – 13:00

Tutorials

- Timeline:
  - Session 1: 8:30 – 9:20
  - Tag team (+ TA break)
  - Session 2: 9:40 – 10:30
  - Tag team (+ TA break)
  - Session 3: 11:00 – 11:50
  - Tag team (+ TA break)
  - Session 4: 12:10 – 13:00

Tutorials

- Tutor's "rights and duties"
  - Tutor WILL
    - Guide you while proposing pseudocode steps
    - Help you with the translation of the pseudocode into Java code
    - Recommend next steps
    - Identify weaknesses and recommend concepts to be revised
  - Tutors WILL NOT
    - Set up Visual Studio Code (Assignment 0 & Office hours)
    - Solve conceptual doubts (Office hours)
    - Help you with code other than the Game (Discord & *Ask a TA* sessions)
    - Help you with Codegrade tests (Discord & Office hours)

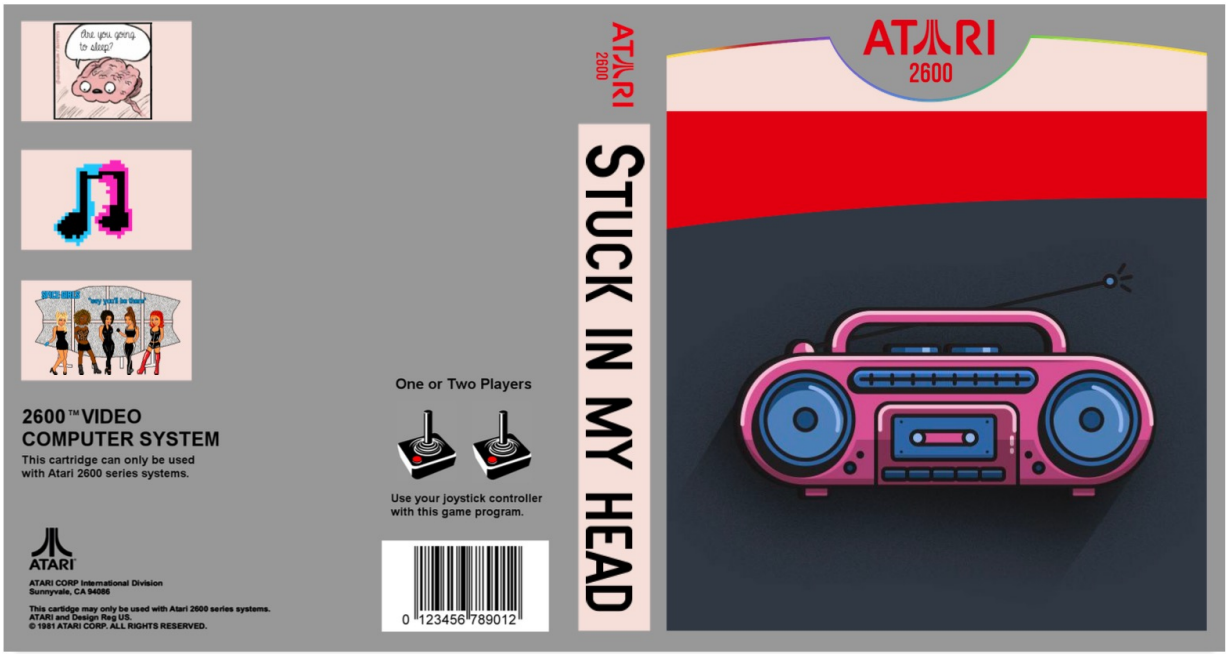
Tutorials

- In the tutorials, we are building a basic version of *Jerry's adventure* game
- Prepare the tutorial BEFORE the session
- We encourage you to work **in pairs/small groups** on the game to improve it **after the tutorial**
- The last session of the course will be an Award ceremony where those who submit a final version will participate
- The best games developers will receive a (small) prize
- More information will be delivered in Canvas soon

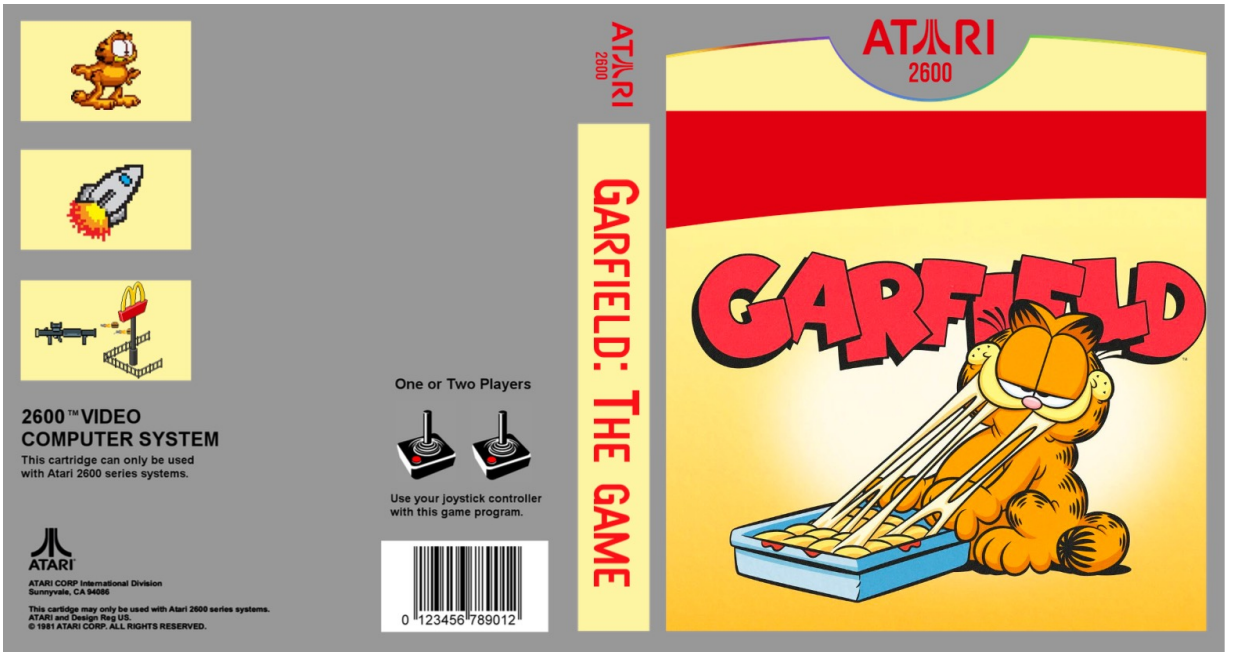
The game – Collecting badges

- 5 (digital) badges available in 4 Labs (Lab 2 is double as it includes 2 tutorials)
- You need a minimum of 4 badges
- Be careful with the last tutorial!
  - Not needed for the game
  - Very limited time due to Project 1.1

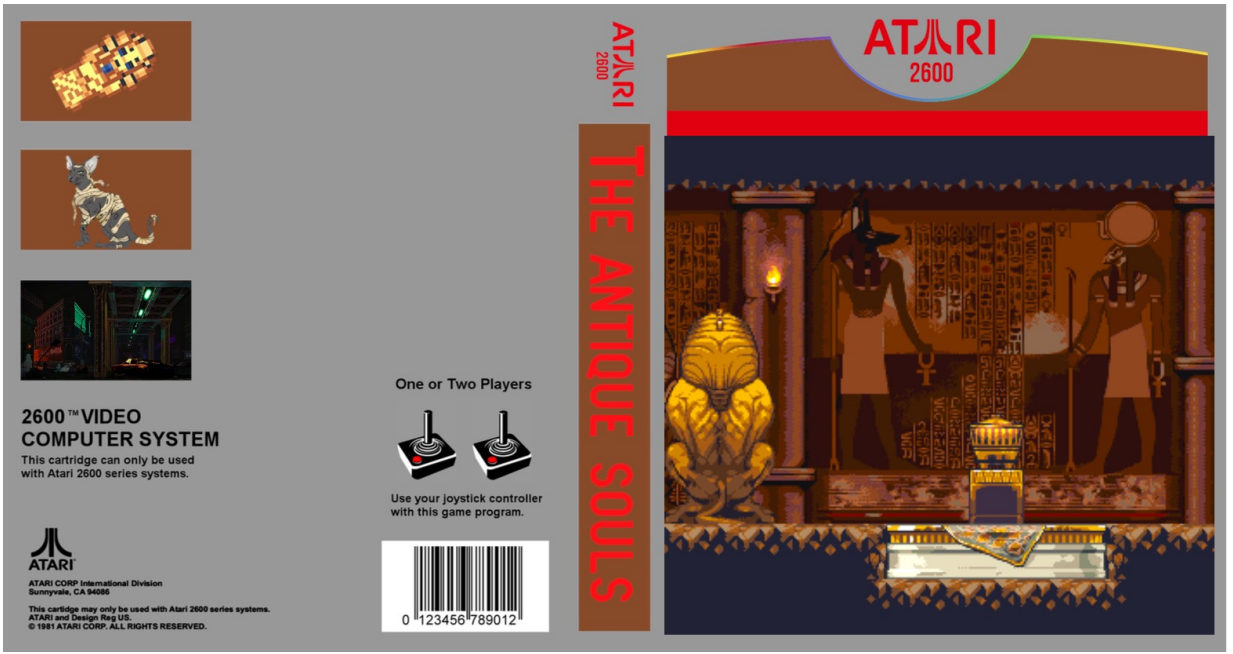
Previous game contest



Previous game contest



Previous game contest



Assignments

- 5+1 proposed assignments
  - *Real* programming problems: receive a (set of) problem(s) and construct a Java solution
- Assignment recommendations:
  - For you to do on your own **INDIVIDUALLY**
  - Validate your implementation automatically (more later)
  - They are all available from the beginning (*Assignments* page in Canvas)
  - They have an estimated due date but you are free to work on them later as well

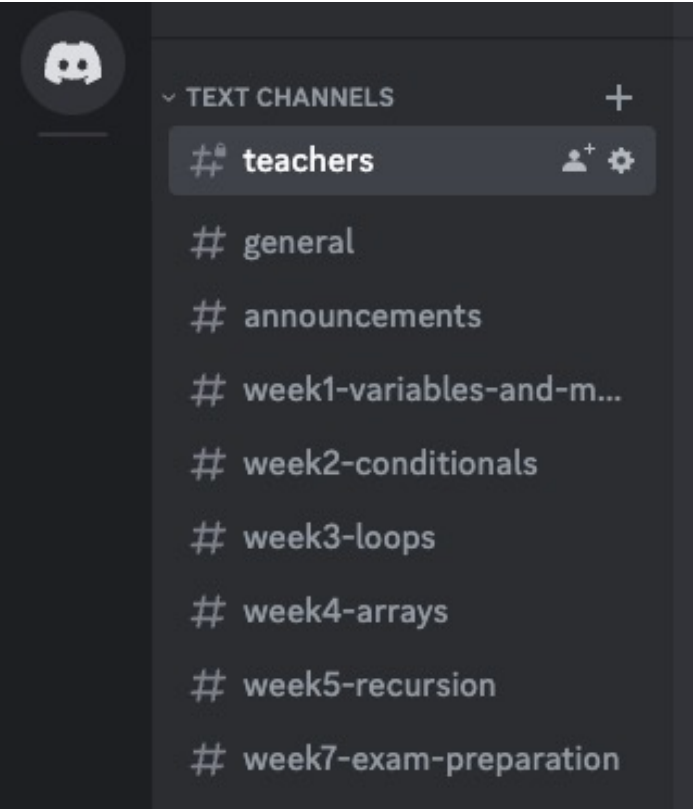
Assignments

- New this academic year
  - Not graded, but a VERY important component
  - Around 20% of the final exam will be closely based on (one of) these assignments
  - The due date is just informative, work at your own pace

Assignment assessment

- You have tests available in CodeGrade
- Code is automatically checked for correctness
  - Several submissions possible (check limits)
  - The tests are just intended as a validation of the code but they do not ensure a perfect behaviour. You must evaluate your implementations.

Discord server



- Join us!
  - https://discord.gg/FFsTsQBfg (only available 30 days, ask for a new link otherwise)

The importance of practicing

Programming can ONLY be learned by doing it!!!

- We can teach you the available concepts and rules, i.e. the grammar and vocabulary
- You have to practice … **really** … you do …

Compare to reading vs writing a book …
or understanding vs speaking a language

Reading a solution is NOT the same as building one yourself from scratch

Software Environments

- The official IDE for this course is Visual Studio Code
  - You can use any other IDE but we (lecturers and TAs) will not solve configuration issues for those
  - You can also use a plain text editor 😉
- Check instructions at Assignment 0
- Check videos for installation etc.
  - Check videos for the use of CMD/terminal

Exam

- Final exam (organized by DACS) is going to be on UM Chromebook (but not with a compiler)
- We will have a mock-up exam session during last week
- Programming is not about learning by heart
- Must get to know certain concepts to allow discussion with other programmers

Thus: Theory + Programming exercises

1. Showing mastery of important terms/concepts
2. Translating tasks/problems into an algorithm
3. Being able to write Java without outside help (or a compiler)

Final grade is earned during 2 hours of very hard work!

Assessment

- What we take into account
  - Participation in all quizzes
  - Knowledge of the assignments (~2 points in the exam)
  - Exam (100%)
- Work now, enjoy later…
  - This course is followed up by:
    - Objects in Programming
    - Project 1.1
    - Data Structures and Algorithms
    - Software Engineering
    - Project 1.2
    - More…

So you think you can code?

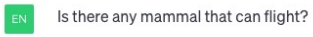
Ok, check it out!!

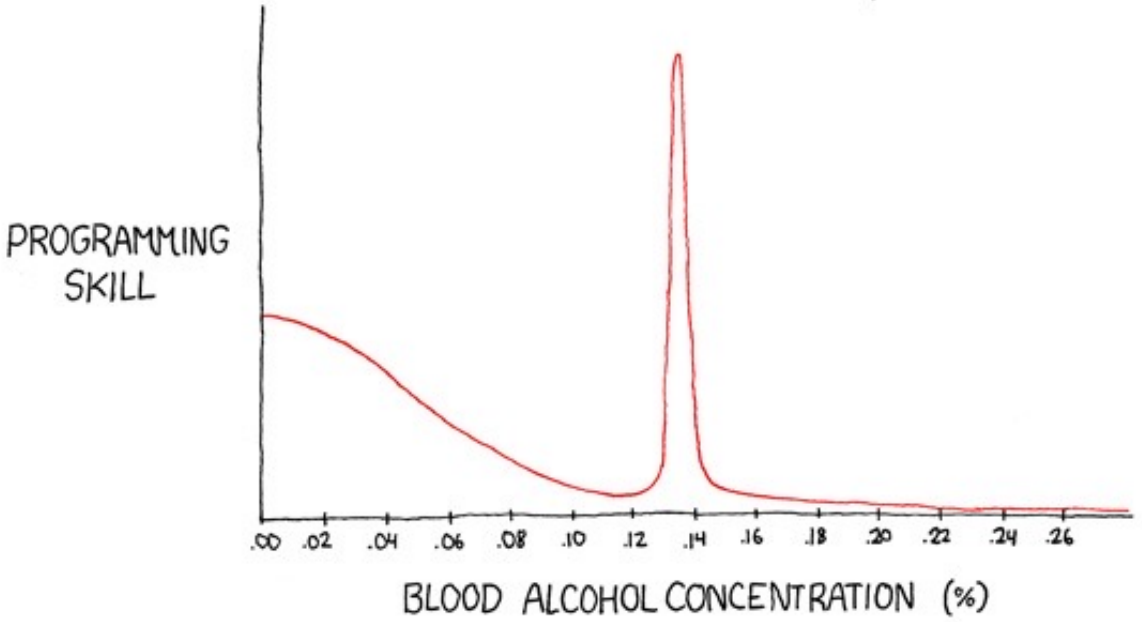
1. All the assignments covering the course Learning Objectives are ready and waiting.
2. Once you complete all assignments, you should be ready.
3. See you at the exam!

Guidelines to survive

- We can show/teach/explain things, but programming is all about ***practice***
- There are several ways the course provides to help you study and understand the material
  - Lectures, Slides, Textbook(s), …
  - Homework, Exercises + assignments
  - Previous year exams (we will give as many as possible)
  - Online material (use WWW wisely)
  - Your teachers and tutors ☺
  - Fellow students
  - LLMs (be careful with them, though)
- Find what works best for ***you*** and use it!

Guidelines to survive



If none of the above works…

- Ballmer peak