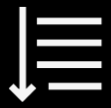
Programming Fundamentals

Review M1-M5 + Prepare the Exam

KdG University of Applied Sciences and Arts

Contents



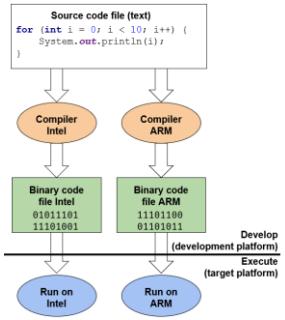
1. Which basic concepts did we see in which week?

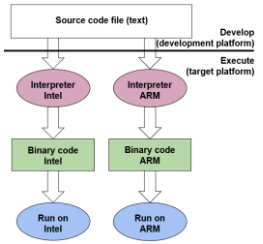
2. How to prepare for the Exam?

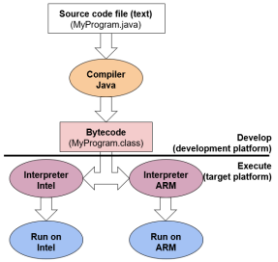
Which basic concepts did we see in which week?

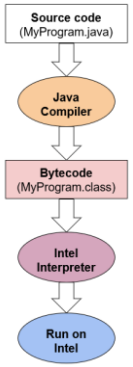
M1 overview

- Programming languages:
  - 3GL, Compile, Interpret, source code, bytecode, binary code,…
- The Java platform
  - JVM, JRE, JDK, Java API, IDE
- Your first Java program
  - HelloWorldApp
- Programming Algorithms









- Algorithm:
  - Sequence of actions
  - Typical actions:
    - Ask Input
    - Produce Output
    - Test
    - Repeat
    - Compute
  - Elaborate actions to the level understood by the computer

M2 overview

- Variables & literals:
  - Variable: location in memory
    - Name (rules & code conventions)
    - Type
    - Scope
  - Varable declaration
  - Variable initialisation
  - Final variable
  - Literals
  - Conversions between types

- Operators:
  - mathematical
  - Relational
  - logical
  - Assignments
  - Conditional
  - other operators
  - operator precedence

short circuit evaluation

Operator
.
[]
(params)
(type)
new
instanceof

Operator
>
>=
<
<=
==
!=

Operator
+=
-=
*=
/=
%=

Operator
+
-
*
/
%
++
++
--
--
-
+

Operator
&&
||
!

Operator
?:

- Increment ++

FIRST increment THEN assign	Pre-increment	Post-increment	FIRST assign THEN increment

	`int number;`
	`int i = 10;`
	`number = ++i;`
	`// Value of number: 11`
	`// Value of i: 11`
		`int number;`
		`int i = 10;`
		`number = i++;`
		`// Value of number: 10`
		`// Value of i: 11`	

- Decrement --

FIRST decrement THEN assign	Pre-decrement	Post-decrement	FIRST assign THEN decrement

	`int number;`
	`int i = 10;`
	`number = --i;`
	`// Value of number: 9`
	`// Value of i: 9`
		`int number;`
		`int i = 10;`
		`number = i--;`
		`// Value of number: 10`
		`// Value of i: 9`	

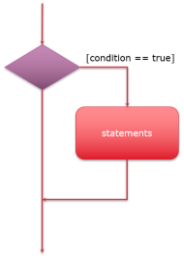
M3 overview

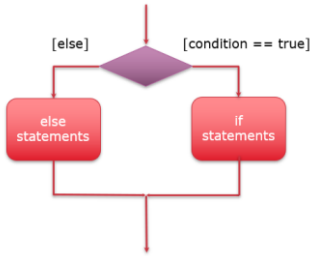
- Expression — *A construct made up of literals, variables, operators, and/or method calls, that evaluates to a single value*
- Statement — *Basic instruction*
- Code block — *A group of statements enclosed within {a pair of curly braces}*

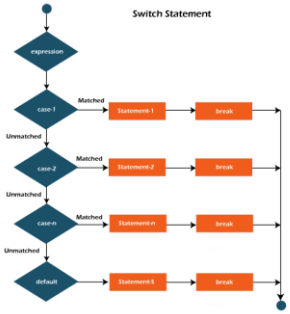
- Flow control statements:
  - Conditional execution:

'old' syntax: fall through

`if` – **`if..else`** – **`switch`**

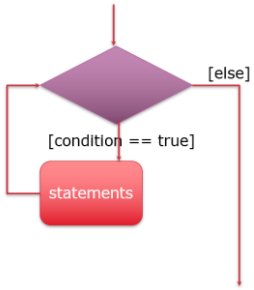


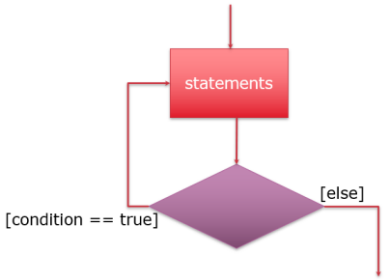


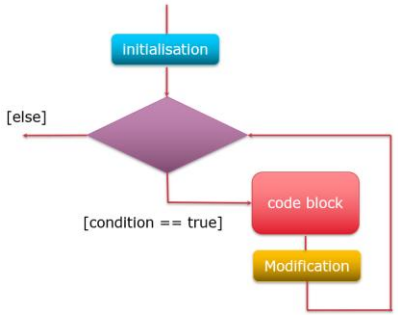


  - Loops:

`while` – **`do..while`** – **`for`**

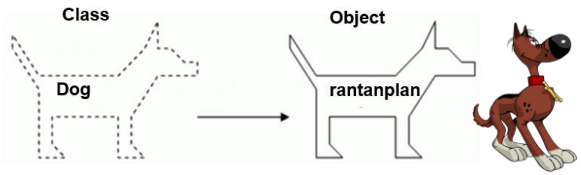






M4 overview

- Object Oriented programming
  - Introduction
    - Class ↔ Object
  - Working with objects
    - Creating objects,
      using and deleting objects
  - Simple objects
    - Classes Random, String & StringBuilder
      - https://docs.oracle.com/javase/8/docs/api/java/util/Random.html
      - http://docs.oracle.com/javase/8/docs/api/java/lang/String.html
      - http://docs.oracle.com/javase/8/docs/api/java/lang/StringBuilder.html
  - Formatting
    - Methods printf & format



Attributes	Methods
String colourPelt;	sit
String colourEyes;	lie
int height;	here
int weight;	

Attribute values	Methods
brown with white spots	sit
dark brown	lie
50 cm	here
35 kg	

```
Dog rantanplan = new Dog("brown with white spots",
                         "dark brown", 50,35);

char position = rantanplan.sit();

Random random = new Random();
int number = random.nextInt();
```

```
System.out.printf("%b %c %4d %.11f %s",
                  true, 'X', 1, 3.14, "abc");
```

Format specifiers (by default aligned to the right):
- `%d` for `int` and `long` (integer number)
- `%f` for `float` and `double` (decimal number)
- `%s` for `String`
- `%c` for `char`
- `%b` for `boolean`
- `%n` for a newline (more platform independent than `\n`)

Extra indicators between % and the type (t, f, s, ...)
- `n` specify the minimum number of positions (i.e. `%20s`)
- `-` (minus) indicates align to the left (i.e. `%-20s`)
- `n.d` when used with `%f` indicates the (total) minimum amount of positions and the amount of decimals (e.g. `%6.2f`), always automatically rounds to nearest
- `,` when used with numbers indicates a thousands separator (e.g. %,d)

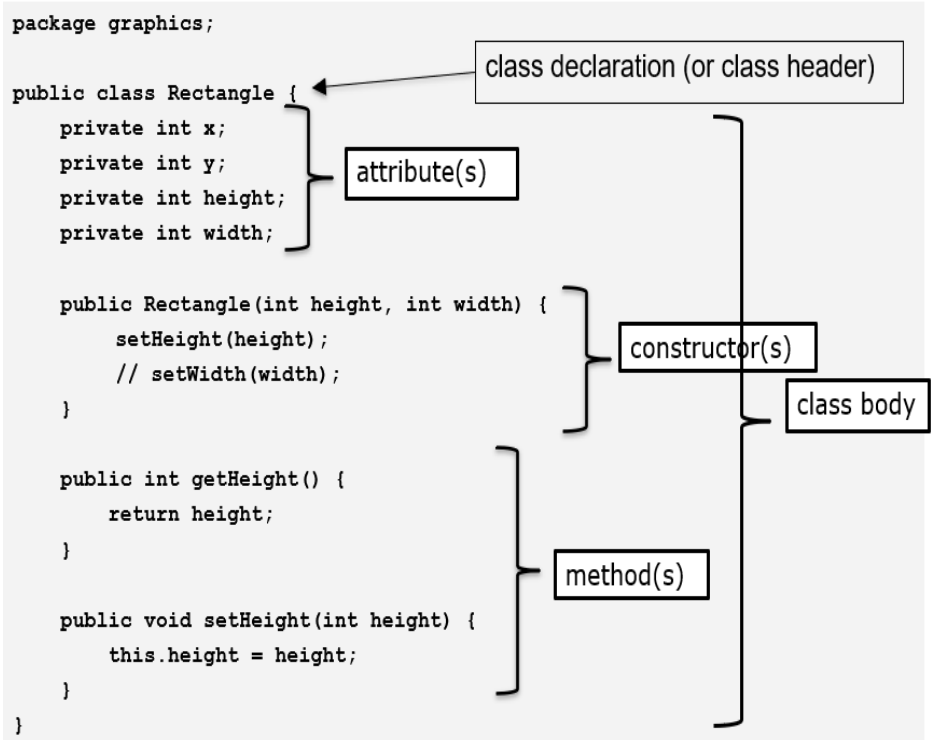
M5 overview

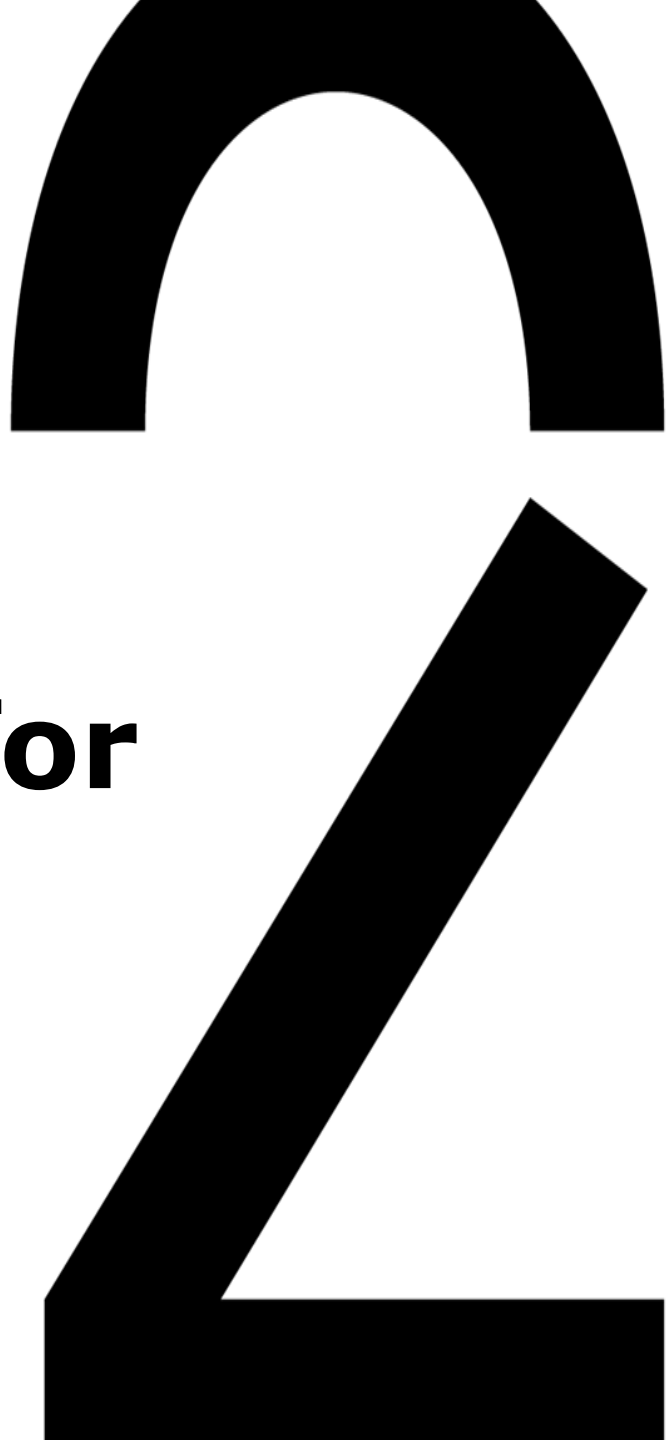
- Class structure
  - Class declaration (aka Class header)
  - Class-body
    - Attributes (Accessibility –encapsulation-, default values)
    - Methods a/o. getters & setters
      (Return type –a/p. void-, local variables, this.nameAttribute, parameters –pass by value-, method overloading)
      toString

```
public String toString() {
    return "(" + x + "," + y + ")";
}
```

```
public String toString() {
    return String.format("(%d,%d)", x, y);
}
```

    - Constructors a/o. default constructor
      (this())
  - The class Math
    - http://docs.oracle.com/javase/8/docs/api/java/lang/Math.html





How to prepare for the Exam?

Organisation of the Exam

1. The exam will be run in a separate canvas course:
   Exam Programming Fundamentals
2. The exam will be an **offline open-book laptop exam**, meaning that you can use everything –EXCEPT AI-tools!- at your disposal on your local machine without using the internet. Make sure you have downloaded them! This includes:
   - **Slides** (you will not be able to use the slides from this canvas course, only those you have downloaded as a PDF on your computer!)
   - **Exercises** you've made
   - Make sure the **java documentation** is available to you **offline**. This way you can look up methods in the Java API and their behaviour during the exam. See Ex 01.02 Run your first program and install Java (Task 3).
   - You may also use paper documentation.
3. Use of bluetooth is forbidden: make sure your computer works without bluetooth (no bluetooth keyboard, mouse...)
4. The use of your internet connection will be monitored by the XMON monitoring tool. Make sure this tool is installed on your laptop.
   It is your responsibility to have the tool installed and running on the day of the exam.

Student instructions Exam

Student instructions

This exam is **monitored by xmon**, the anti-fraud tool of Applied Computer Science. The tool checks whether or not you make use of the internet or other tools (such as AI) during the exam.
Close **all** applications that you do not need during the exam!
After the exam has started, xmon should show a green frame or icon for the duration of the exam.

You will need a **CITRIX code to start the tool**. This code is to be provided by your supervisor at the beginning of the exam.

Read and follow the instructions provided here, in the Canvas Quiz and in xmon.

Download the IntelliJ project with the **start code** before disconnecting your network/wifi/Bluetooth. You can follow the instructions below:

1. **Download and Unzip**:
   - Download the starter project zip file from Canvas.
   - Unzip the file to a location on your computer where you can easily access it.
2. **Open the Project in IntelliJ IDEA**:
   - Launch **IntelliJ IDEA** on your computer.
   - Go to **File** -> **Open** in the IntelliJ menu bar.
   - Navigate to the folder where you unzipped the starter project and select it. Click **Open**.
3. **Verify Project Setup**:
   - Ensure the project opens without errors.
   - If prompted, make sure you select **IntelliJ** as a build system for the project.
   - Check that:
     - Your **JDK** is set up correctly. (Go to **File** -> **Project Structure** -> **SDKs**. Ensure the JDK version matches the required version in the assignment.)
     - All dependencies are downloaded successfully.
4. **Run the Starter Code**:
   - Locate the main class in the right module (e.g. **Q1**) of the project (e.g., **Main.java**).
   - Right-click the file and select **Run Main** (or the equivalent for your IDE version).
   - Verify that the starter code runs without errors and produces the expected output -if any-.
5. **Ask for Help if Needed**:
   - If you encounter issues during any of the steps, ask your instructor or TA for help before proceeding with the assignment.

Be sure to **upload** at the end of the exam your zipped **IntelliJ project with your answers**, NOT the initial start code! You can follow the instructions below:

- Only once you are done with the exam and/or instructed to do so.
- **Zip** the content of the project directory again.
- Make sure to **name** your zip file as ∫. Example: **lastname_firstname_ACS10x_progfun_T1.zip**
- **Upload** the zip file into canvas

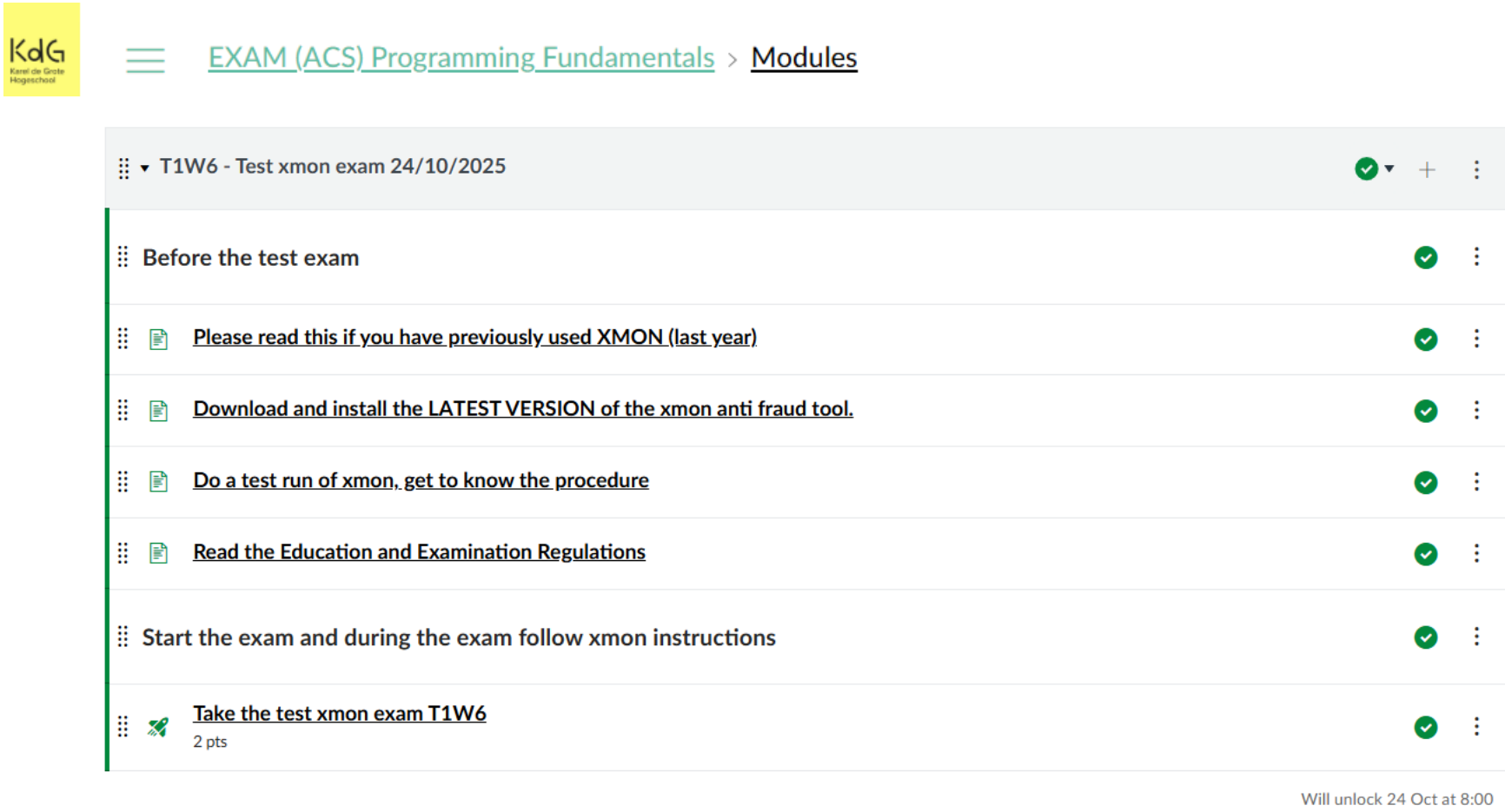
Also upload at the end the **xmon log file** (xmon shows on its final screen the exact location) in Canvas. **Without** the xmon log file your **exam submission is not valid!**

IMPORTANT REMARK
Apply the correct Java object oriented and other concepts according to Java-conventions, Java best practices and programming techniques demonstrated/used in class! Failing to do so will impact your grade even if your code can be executed and does deliver the requested output.

TIP:
Do not waste all of your time focusing on one question. If you get stuck for more than 15 minutes, see if you can finish any other of the questions first.

How to prepare for the exam?

Familiarize yourself with the exam format:



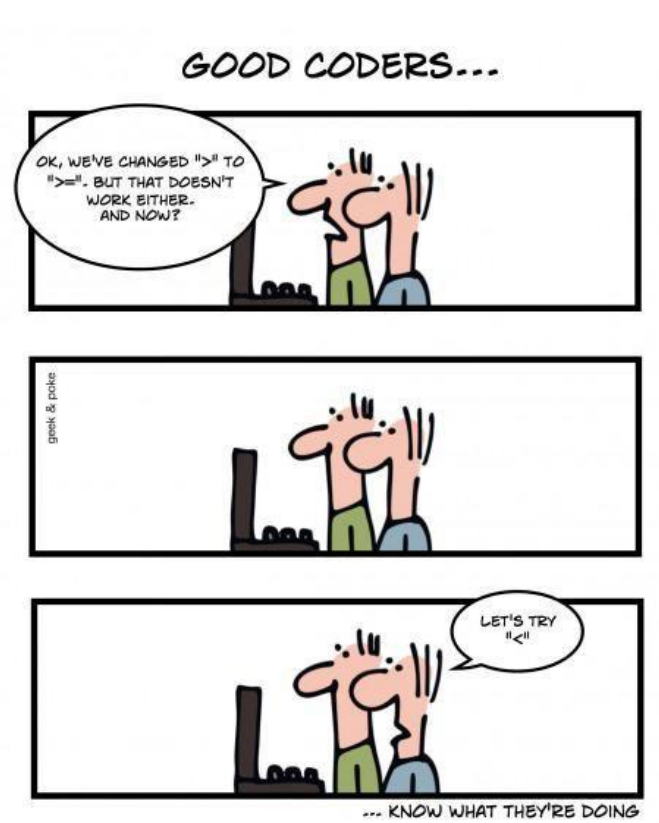
How to prepare for the exam?

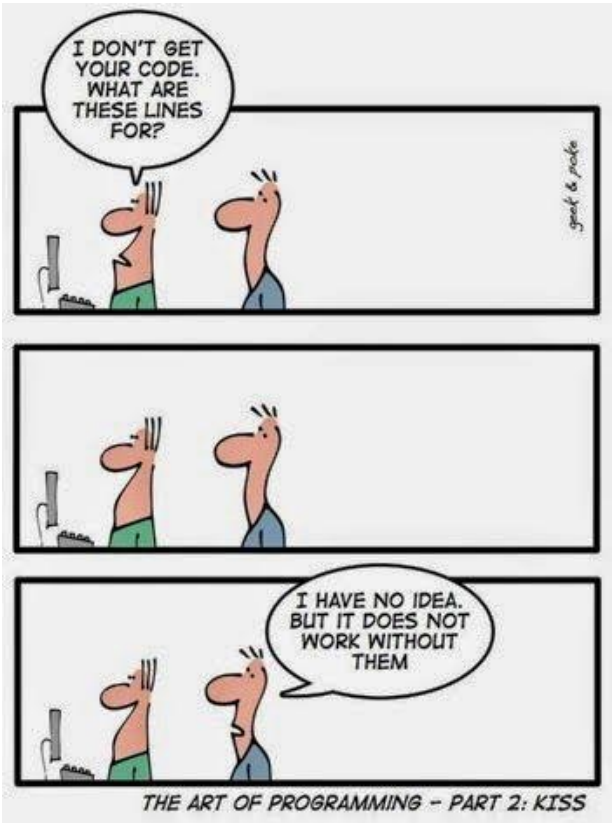
Make a lot of exercises (see Canvas and slides)

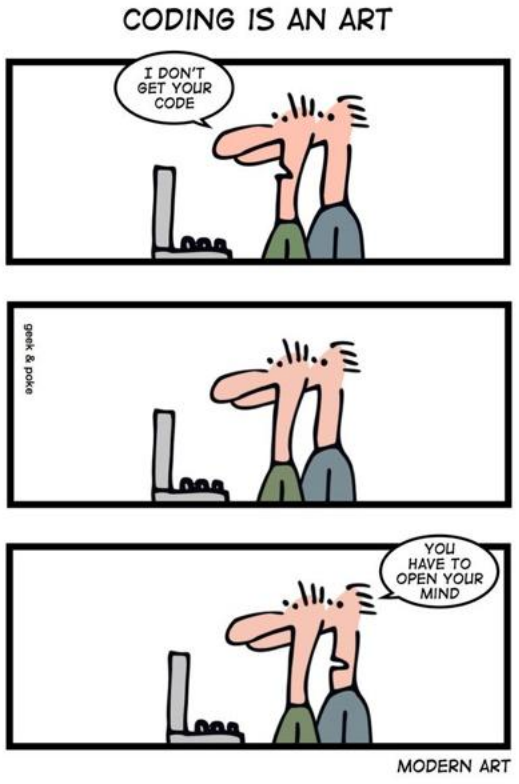
→ Make for each topic at least one exercise
(and more until you got in in your fingers)

Look at

- The brownfield challenges
- The extra exercises







How to prepare for the exam?