AARHUS UNIVERSITET

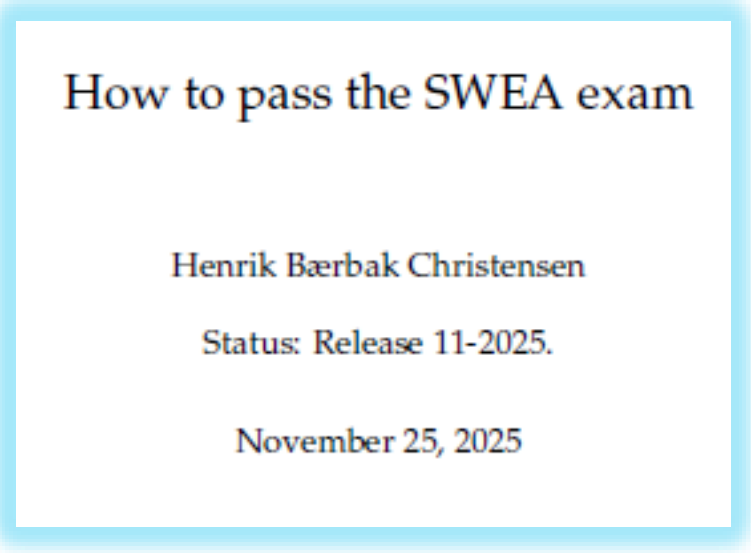
Software Engineering and Architecture

The Exam

AARHUS UNIVERSITET

Exam guide

- Release Nov 2025!
  - Feedback is very welcome
- On the last week plan
- *Best: Rehearse during TA classes !*
  - Some of you have already done so!


How to pass the SWEA exam

Henrik Bærbak Christensen

Status: Release 11-2025.

November 25, 2025

AARHUS UNIVERSITET

When

- A few before Christmas, the rest in January
  - Censors:
    - Clemens Klokmose,
    - Ole Caprani
    - Kasper Green
    - Jaco van de Pol

- The examination plan is on BrightSpace, and will be updated once ‘mandatories’ have been settled
  - *The faculty one is out-of-date!*

AARHUS UNIVERSITET

The Execution

- Examination format: **Physical + Zoom** based
- Physical
  - You get you exam exercise **on paper.**
  - The exam is physical in the sense
    - In the same room with me and censor
    - We talk ☺
- Zoom based
  - You present your solution produced during the preparation time using *Zoom shared screen*
    - *Code stuff in particular…*
  - And of course the whiteboard if you wish (and paper)

0.2 Test-driven development.

Consider the following specification:

```
1  public interface FanControl
2    /** Return the frequency of the cooling fan given the
3        temperature of air and liquid in a chemical chamber.
4        The ideal liquid temperature is around 75 degrees.
5
6        The frequency (return value) is calculated as follows
7        (in order of precedence):
8
9        if TempLiquid > 90 return 9999 (ALERT)
10       if TempLiquid > 80 return 500 (Max cooling)
11       if TempLiquid < 70 return 0 (No cooling)
12       otherwise return (TempLiquid-70)*50
13
14       The air temperature may overrule the above calculation:
15       if TempAir > 100 return 9999
16       if TempAir > 90 return 500
17     */
18   public int fanControl(double TempAir, double TempLiquid);
19 }
```

You are asked to describe a preliminary plan for a test-driven development effort. You should use terminology, techniques, and tools from the course to:

- Sketch a test list, and outline some plausible initial iterations.
- Cover steps and TDD principles in one or two initial interations, time permitting, including central Java code fragments.
- Broaden the discussion to include basic definitions, terminology, and techniques in the area.
- Relate to other topics in the course.

Do not try to cover all requirements if this removes the possibility of broadening the discussion.

How

- Be in the **waiting room of exam Zoom session**
  - Best before your exam actually starts ☺
- Have your ***study id card*** ready for verification
- Draw a ***random exercise***
  - *The exercise is a single A4 physical piece of paper*
- **Read, understand, and 'start solution'**
- I do not expect a complete and polished solution!
- *Better to understand the exercise than show a solution to something else than the exercise*
  - Not a disaster, but it takes valuable time out of the examination

AARHUS UNIVERSITET

The Exam Exercise

- It should be obvious, but still...

Any form of copying and distribution of an exam exercise is exam cheating. Exam exercises are secrets.
Any instances of copies on any platform should be reported to me!
After exam, you must delete produced material.

AARHUS UNIVERSITET

Use of CoPilot/ChatGPT

- Repeating what has already been said a few times

Use of AI tools is permitted, but must

a) Be stated up-front : what have you done?

b) Is considered "other's work" and does thus *not count as anything you have done!*

An exam showing a correct solution created by ChatGPT will of course lead to grade -3; as SWEA is not a prompt engineering course!

AARHUS UNIVERSITET

How: The Preparation

- During the preparation period (30 – 35 min)
- **Use your own laptop to overall ‘solve’ (parts of) the exercise**
  - Prepare a UML diagram
  - Prepare a EC and test case table
  - Prepare code skeleton / partial solution
  - To be presented at the exam…
- **Resources available: “Everything” except any help from 3rd party**
  - No help from other persons; use of AI tools considered ‘other’s work’

AARHUS UNIVERSITET

The Exam

- We sit together – we talk and discuss
- Censor, I and you are on Zoom and you **share your screen**

- Present your *developed code/material* and talk us through what you have made – and we ask questions etc.
  - And ask you to change and add stuff…

- 11-12 minutes only
  - ‘but … I only have started’

AARHUS UNIVERSITET

After Exam

- After the exam we will
  - Of course, give you a grade, explain the reasoning behind it, and provide advice wrt. future exams
  - **Require you to delete produced material**
    - As any distribution of your developed material and the exercise itself is considered *exam cheating.*

AARHUS UNIVERSITET

Grading

- Starting E2022 the mandatory points are *not counting towards the final grade…*

- **Mandatory point 60% of 440 = 264**
  - **Handin of all 10 exercises**
  - Is required to attend the exam

- Final grade is based upon oral exam *only.*

AARHUS UNIVERSITET

Some Stuff You May Meet

AARHUS UNIVERSITET

Realistic Exam Exercises

- All exam exercises are modelled after realistic systems!
  - No exercises ala
    - Use Broker to implement objectFoo.doSomething(27,”Fisk”);
- Which means you *will run into Java classes like*
  - ZonedDateTime

```
ZonedDateTime time = ZonedDateTime.now(); // Get Current time and date
```

  - PrintStream

```
private PrintStream logfile;
logfile.println("
```

- And code inspired by HotStone, MiniDraw, TeleMed, PayStation…

AARHUS UNIVERSITET

Realistic Exam Exercises

- All exam exercises are modelled after realistic systems!
  - … which means it will contain realistic terms from IT and domain
- I expect you to know what terms like
  - GPS, Mobile phone, app, SMS/Text message, database, caching, server, blood pressure, inventory,
- … means
  - Often terms are explained or written in Danish
    - Tenant (Da: “lejer af lejlighed”)
    - Caching (That is, store a local copy of data so a remote fetch is not required)

AARHUS UNIVERSITET

Hints and Advice

Format

- Use the provided exam question set to ***rehearse*** the exam format
  - Pick an exercise from the example set
  - Spend 20-25 minutes
    - Finding solution techniques
    - Partially solving it **using your laptop with appropriate tools**
  - Spend 12 minutes doing a ‘dry-run’ exam with your group
    - Let your other groups members act examiner
  - Rotate and repeat ☺

AARHUS UNIVERSITET

Solve the Exercise

- SWEA is not just defining three interfaces with a method in each…
  - I see a lot of ‘solutions’ at the exam of this type
- **Where are they used? Where are those methods called? By Whom? How were they created?**
  - Ala ‘theObject.doSomething(….)’

AARHUS UNIVERSITET

Appropriate Tools

- Experience from previous year's Zoom based exams

- **Word is not a good editor for code !!! !!! !!!**
- **Free-hand drawing on a laptop does not work!**
  - A mouse is not a pen. Pen = fine muscles; mouse = coarse…

- Use IntelliJ?
  - Be sure you have tried it out before you do. Do not let it spoil the exam because it insists your code does not compile

AARHUS UNIVERSITET

Appropriate Tools

- Find a *good code editor (that is not an IDE ?)*
  - *Gedit, Emacs, SublimeText, EditPlus, Nodepad++, etc.*
  - It does not choke on pseudo code but knows indentation, syntax highlight etc.
- Find a good spreadsheet type program
  - For making EC tables and test case tables
  - Filling out the Clean code analysis
- Find a decent way of drawing UML
  - An editor – or – draw it on the whiteboard at exam
  - Online tool is ok!

AARHUS UNIVERSITET

Test the Technique

- Zoom Share Screen is essential and must be tested
  - Mac's require a setting to be changed and the machine rebooted
    - Ok, so the first 4 minutes of your exam is spent on that
      - **Bad grade due to not testing that aspect in advance?**
  - Screen sharing does not work on certain Ubuntu versions!
    - **Bad grade due to not testing that aspect in advance?**
  - MacOS require a permission and a reboot
    - Do it beforehand, do not waste time at the exam
      - Bad grade….

And…

Increase the font size! ☺

AARHUS UNIVERSITET

Right?

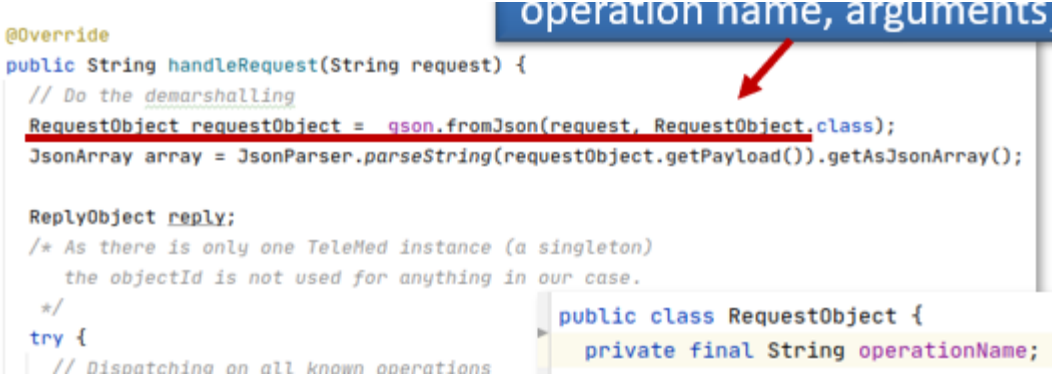
- *“As you can obviously see in line 17, I have coded the State pattern”*

- *(No, I cannot)*

AARHUS UNIVERSITET

Templating

- ***At the exam you are only allowed to bring whatever you produce in the preparation.***
  - **But you are expected to bring that to the exam!**
- The *exception* to this rule
  - The Invoker code base involves boilerplate code that is the same always (demarshalling 'request' into its request object)
  - It is OK to copy that from a template you prepare in advance…



Nervous?

- Some students are not nervous at exams.
  - How do you do that???
- Some are...
  - Learn to **use** your nervousness, instead of trying to avoid it...
- *Perhaps it is not a problem, it is a condition of life…*
  - *Problems you solve, conditions you learn to live with…*

AARHUS UNIVERSITET

Paths to Failure

Common failing reasons

AARHUS UNIVERSITET

Observations

- Commonalities in failing the SWEA exam
- Level 1
  - Coding mastery – **you are expected to write decent Java!!!**
    - SWEA is a **programming course** after all!
      - Quite a few fail simply because they cannot write a ‘new’ statement in Java ☹
- Level 2
  - Shallow learning / ”parrot knowledge”
  - Reading a specification causes problems
  - A few in the classic category: read the book also…

Coding

- Too many do not fail SWEA, but fail to have sufficient *coding practice* at my exam
  - Writing 'nonsense' and cannot see the problem

```
assertThat( pws.parse(String line), is(true) );
```

```
private ParseWorkSpecification pws;

@Before
public void setup() {
  ParseWorkSpecification pws = new ParseWorkSpecification();
}

@Test
public void shouldDoSomething() {
  assertThat(pws.parse("Wed 1"), is(true));
}
```

What is a constructor?

Horrible statistics

- E2024 about 12% of SWEA students failed in IntProg curriculum
  - Not in SWEA curriculum…

- You to demonstrate **ability to program**, in order to demonstrate ability to do *advanced programming*

Parroting 1

- ”Parrot” knowledge
  - You can learn the *visual appearance* of Strategy pattern UML
  - You can learn the *visual appearance* of Strategy code template
  - You can learn the *common names* used in Strategy

  - Without any clue of what Strategy **is**

- Beware of it!

AARHUS UNIVERSITET

Pattern Matching

- Similar – ”pattern matching” shallow learning
  - Make ECs correctly for an interval in the exercise
    - But fail to mention what heuristics that has been used (range)
    - Fail to know how the range heuristics is formulated
    - Fail to know what an EC is at all
  - Make ‘ECs’ that are test cases
    - 7 [b1]
- Conclusion
  - Learned the ”template”, then put arbitrary stuff into it ☹

AARHUS UNIVERSITET

Exercise Reading

- You will have to read the exercise and understand at least a bit of it

- **Read the code fragments / interfaces – they are the key question**

AARHUS UNIVERSITET

And of course the usual

- A few fail of the same reason as in other courses
  - Have we read the same book and coded the same exercises???

Conclusion

- You have to master *some* of this course to pass it
  - You need some level of mastery of Java or OO language
    - Which you learn by *doing it*
  - You need to understand what is *behind* the templates
    - Which you learn by *doing it*
  - You need to be able to read a small specification
    - Which you do by *reading it*

AARHUS UNIVERSITET

SWEA 2025 signing off…

It has been a pleasure flying with you…