ITP2200
Introduction to Software Testing

Bogdan Marculescu

Høyskolen Kristiania

Lecture 5
Identification of Correct Outputs
- The Oracle Problem -

So, more programming this week?

Høyskolen Kristiania

What do we remember from last time?

Høyskolen Kristiania

We mentioned “Coverage”

```java
public class TriangleClassifier {
    public static String classify(int a, int b, int c) {

        if (a <= 0 || b <= 0 || c <= 0) {
            return "NOT_A_TRIANGLE";
        }

        if (a == b && b == c) {
            return "EQUILATERAL";
        }

        int max = Math.max(a, Math.max(b, c));

        if ((max == a && max - b - c >= 0) ||
                (max == b && max - a - c >= 0) ||
                (max == c && max - a - b >= 0)) {
            return "NOT_A_TRIANGLE";
        }

        if (a == b || b == c || a == c) {
            return "ISOSCELES";
        } else {
            return "SCALENE";
        }
    }
}
```

```java
@Test
public void testScalene() throws Exception {
    int a = 5;
    int b = 7;
    int c = 9;

    String result = TriangleClassifier.classify(a, b, c);
    assertTrue(result.equalsIgnoreCase( anotherString: "SCALENE"));

}

@Test
public void testIsosceles() throws Exception{
    int a = 5;
    int b = 5;
    int c = 7;

    String result = TriangleClassifier.classify(a, b, c);
    assertTrue(result.equalsIgnoreCase( anotherString: "ISOSCELES"));
    assertFalse(result.equalsIgnoreCase( anotherString: "SCALENE"));
}
```

Code is at
https://github.com/bogdanmarculescu/itp2200

Høyskolen Kristiania

Equivalence Classes

Høyskolen Kristiania

Defining ”correct” behaviour - Oracles

Høyskolen Kristiania

Running Examples

- **Division** – simply division of 2 numbers (slightly, modified version of the basic division
- **Sinus** - computes sin(input)
- **Exp** - computes e^x
- **sortArray** – a simple sort of an array of doubles

```
10  @   public static double sinus(double input){
11          return Math.sin(input);
12      }
13      /**
14       * A simple [division] function.
15       */
16
17  @   public static double division(double numerator, double denominator)
18          if (denominator == 0) throw new IllegalArgumentException();
19          return numerator/denominator;
20      }
21
22      /**
23       * Computing e^input
24       */
25      public static double exp(double input){
26          return Math.exp(input);
27      }
28
29      /**
30       * Simple array sorting
31       */
32  @   public static double[] sortArray(double[] array){
```

Høyskolen Kristiania

Test oracles - Implicit

Some behaviours are almost always faults:

- Crashes
- Segfault
- Buffer overflows
- Deadlocks

Høyskolen Kristiania

Test oracles - Implicit

Benefits:

- Clear
- Reasonably general

Drawbacks:

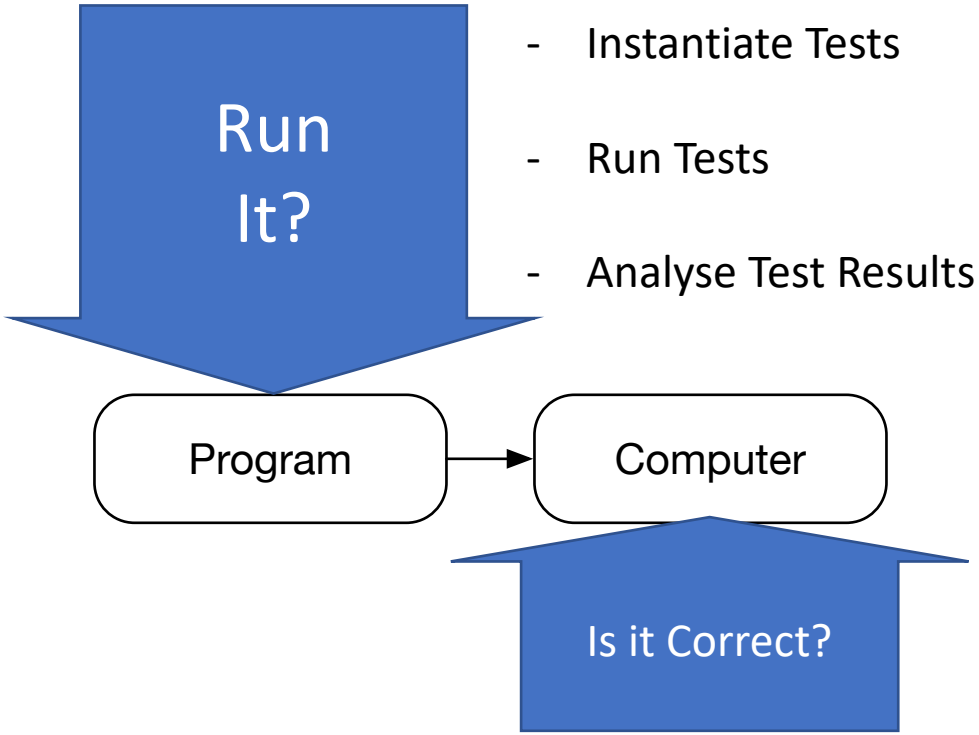
- Not very expressive
- No error does not mean no faults

Høyskolen Kristiania

Test oracles - Human

Remember Lecture 1:

- Design Tests
- Instantiate Tests
- Run Tests
- Analyse Test Results



Høyskolen Kristiania

Test oracles - Human

Human evaluation of the result:

- Great is some cases (console display, webpage design and GUI testing)

- Limits on amount of data

- So, what is
  - sin(42)
  - e^(42)?

Høyskolen Kristiania

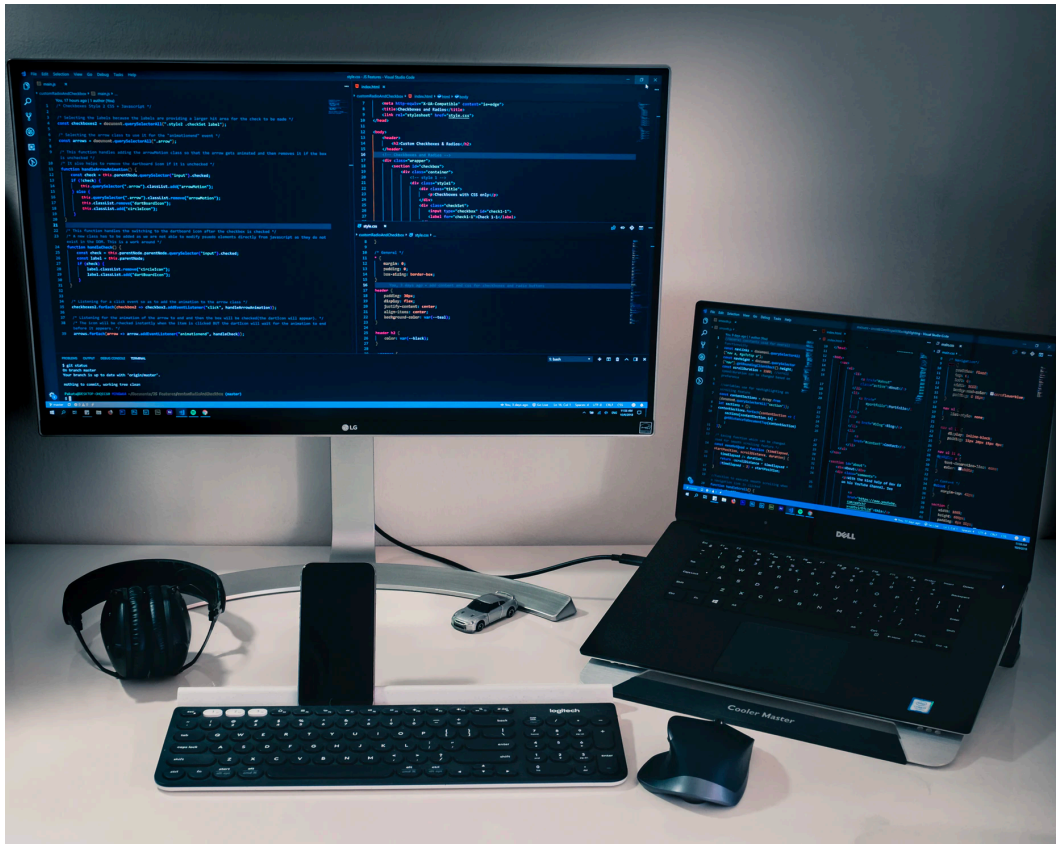
Test oracles - Human

Benefits:
- Expressive
- Easy to use
- Allows for subjective evaluations (design, GUI, etc.)

Drawbacks:
- Costly
- Subjectivity as a drawback
- Accurate computation

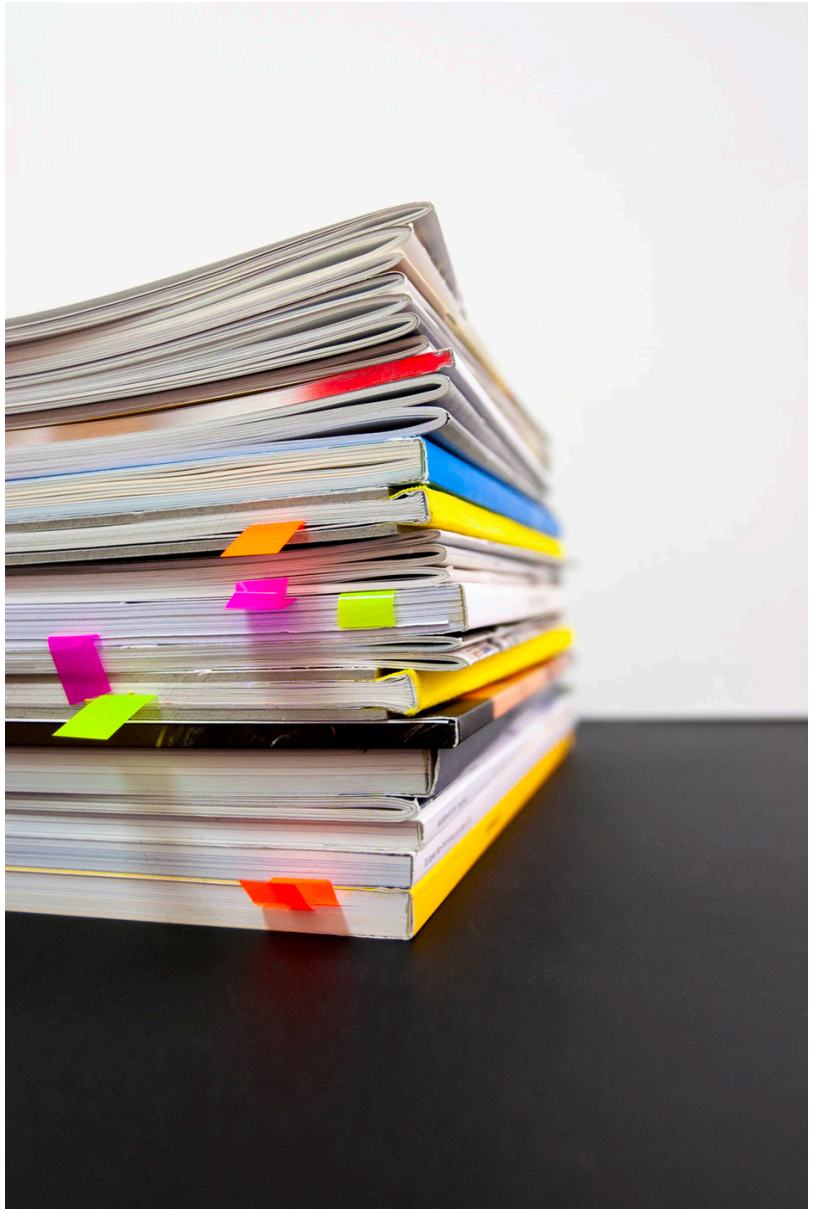
Høyskolen Kristiania

Coding time!



Høyskolen Kristiania

Test oracles - Specified



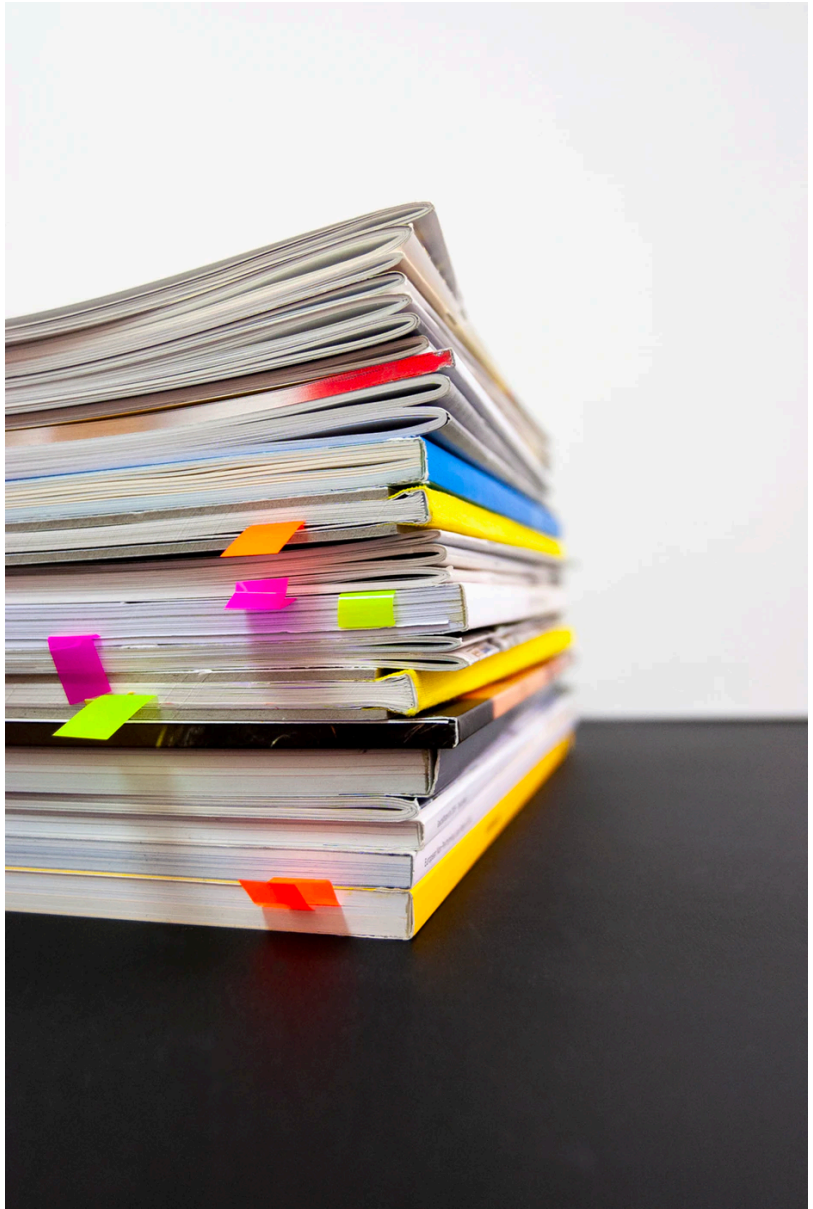
Specification – Software Requirements Specification

A document that describes the software

- Special languages -
  - Models (as in UML, not miniatures)
  - Contracts
  - Algebraic descriptions
  - State machines

Høyskolen Kristiania

Test oracles - Specified



Benefits:

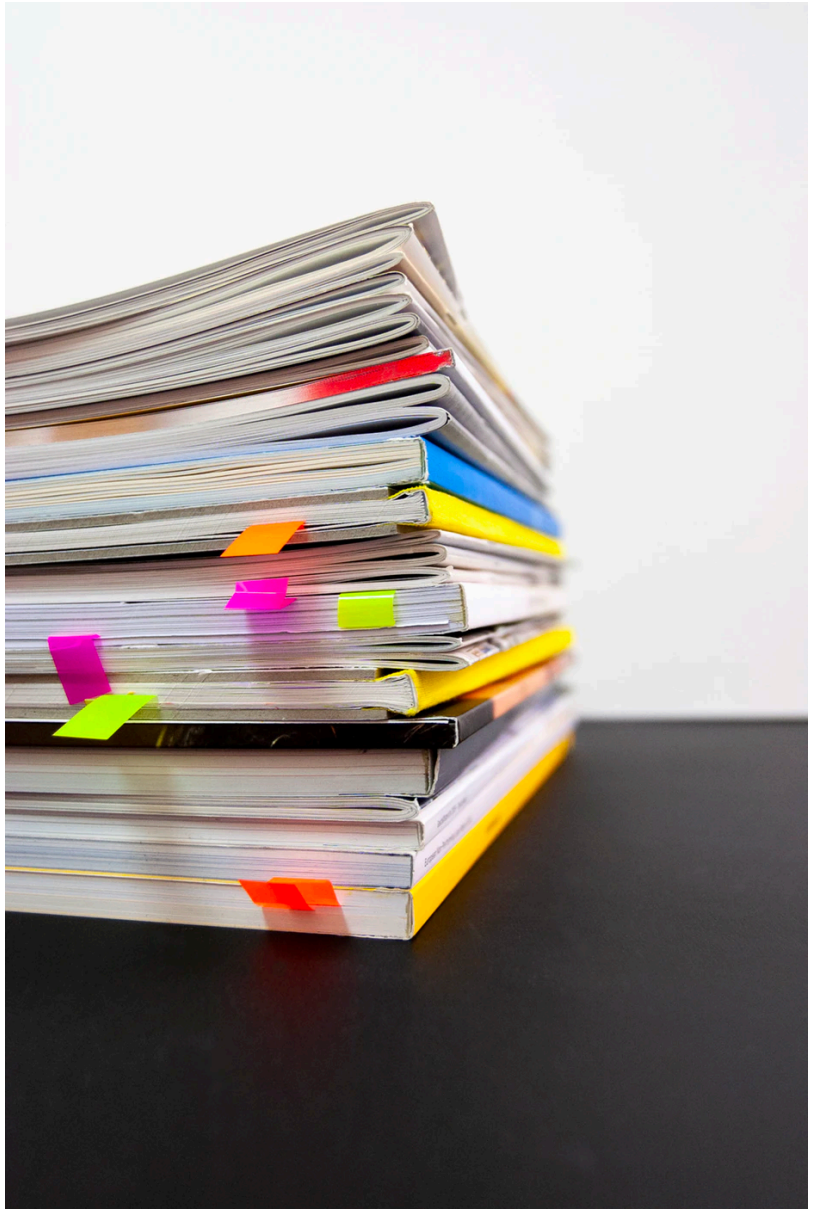
- Can be very **specific**
- Can be very **accurate**
- Can be assessed **automatically** (remember the assertTrue() function)

Drawbacks:

- Requires detailed knowledge of the system under test (SUT)
- Time, effort to develop and maintain

Høyskolen Kristiania

Test oracles - Specified



So!

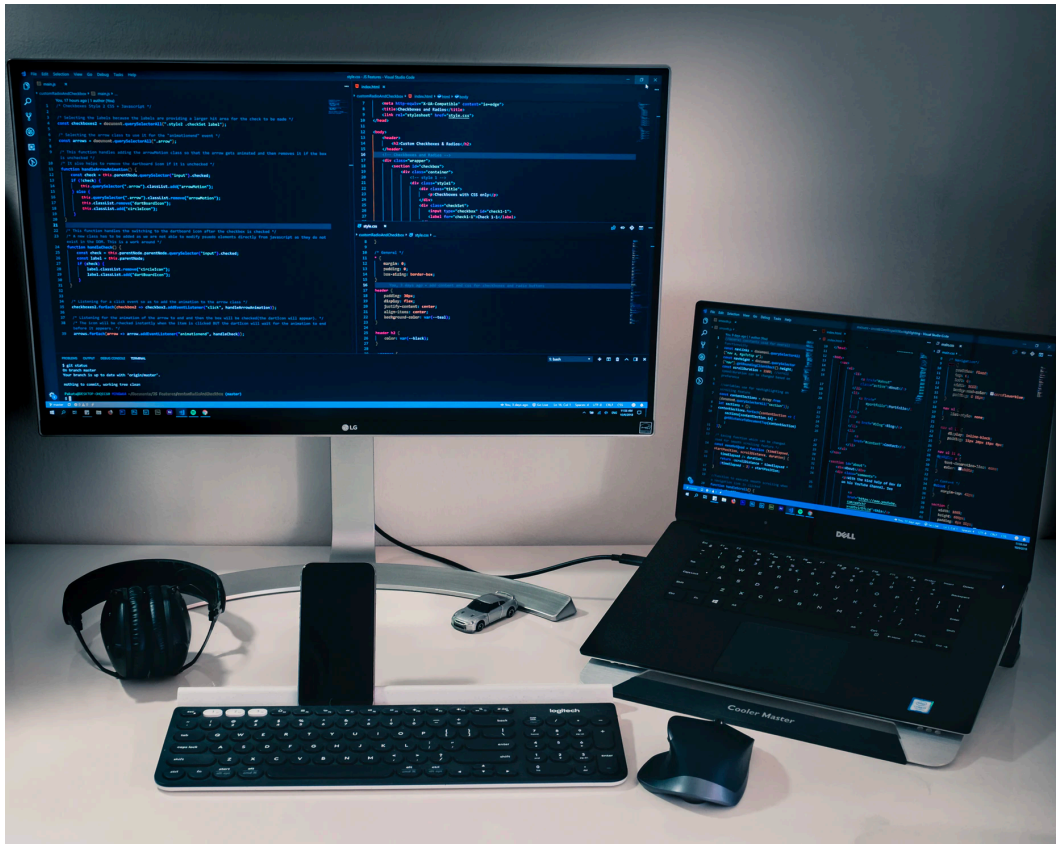
public static double[] sortArray(**double**[] array)

- What **specifications** would be appropriate?
- What can we **assert** about this system?

- Is a complete set of specifications for this possible?

Høyskolen Kristiania

Coding time!



Høyskolen Kristiania

Test oracles - Derived

Derive from other sources:

- Documentation (specification mining, documentation, conversions to specs)

- System execution (invariants)

- Known (or desired) properties (metamorphic testing)

- Other versions
  - Regression testing
  - Other implementations

Høyskolen Kristiania

Test oracles - Derived

Let’s consider:

- sin(x)

- exp(x) – e^x

We cannot (necessarily) calculate or duplicate them.

So, what can we assert?

Høyskolen Kristiania

Test oracles - Derived

Different implementations:

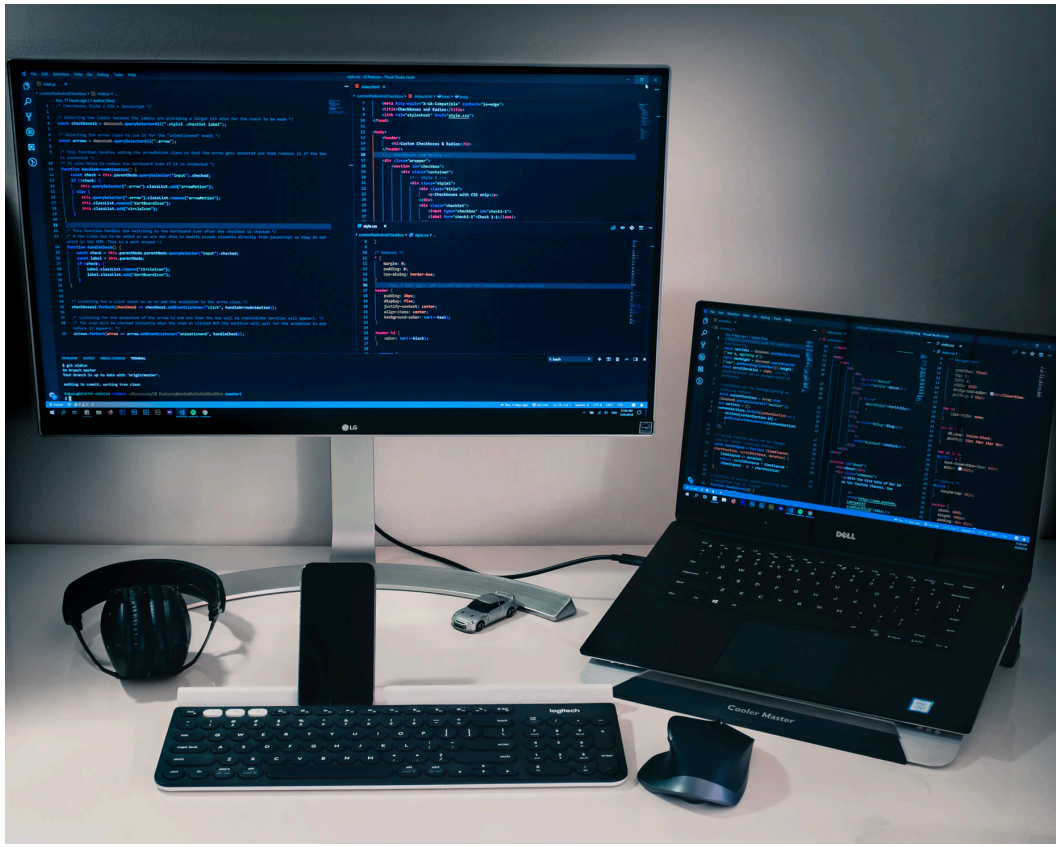
- (remember last seminar)
- Different teams?
- Different versions?

Regression testing

- Assess the behaviour of the **current** version against the **previous** version
- **Automation** is preferable
- Automated asserts, automated evaluation

Høyskolen Kristiania

Coding time!



Høyskolen Kristiania

Test oracles - Derived

Benefits:

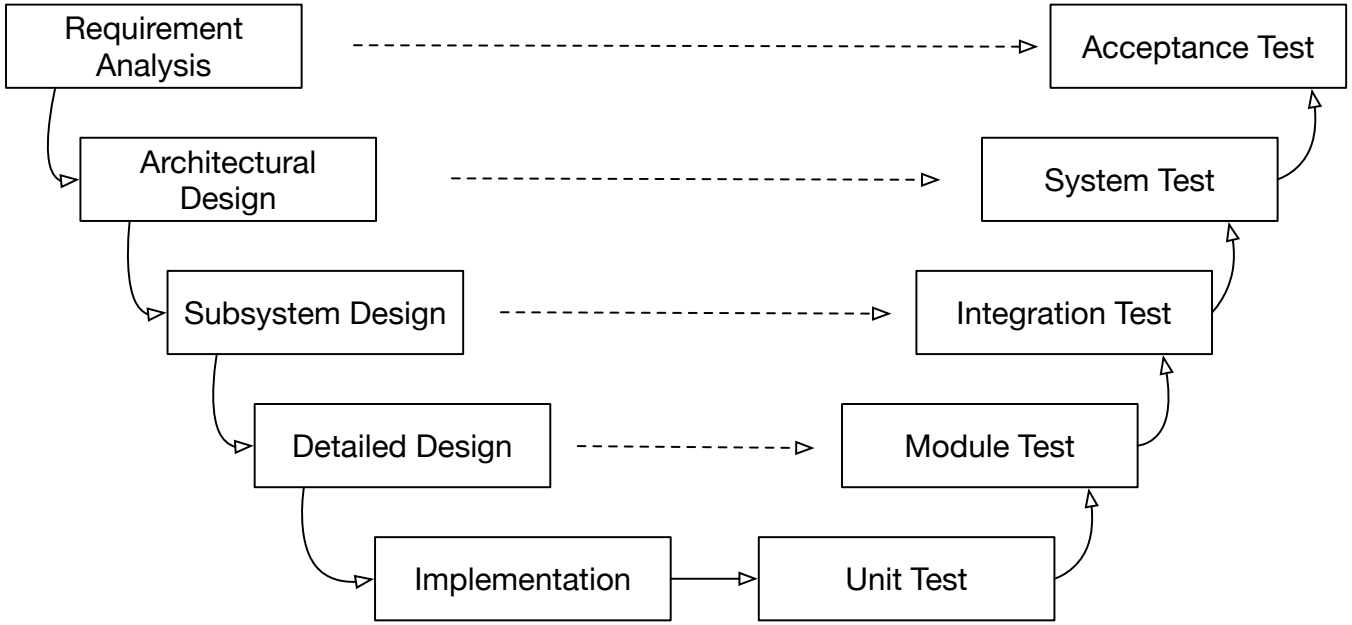
- No need for completeness
- Can be derived from existing artefacts
- Can focus on important properties
- Can be **automated**

Drawbacks:

- Some knowledge required
- Not complete
- Faults may complement each other

Høyskolen Kristiania

Remember this?



- Or the one-sentence guide to the entire field of software engineering

Høyskolen Kristiania

Høyskolen Kristiania

Questions so far?

Exercise for the Seminar

Rooms:

-A3-01

-A4-03

-A5-18

Høyskolen Kristiania

Exercise for the Seminar

Write tests for the Class Time (from OOP):

-convertSeconds(int sec)

- What assertions would you add?
- Write tests for it

Høyskolen Kristiania

Exercise for the Seminar

Write tests for the Class Date (from OOP):

- dayDiff(Date d1, Date d2)
- daysSince0(Date d)

- What assertions would you add?

- Write tests for it

Høyskolen Kristiania

Exercise for the Seminar

If possible, use your OOP implementation

Høyskolen Kristiania