

# **Compulsory Assignment 1**

## **Software Testing**

### **INF3121 / INF4121**

---

**Assignment start date:** 17<sup>th</sup> of March 2017, 15:00

**Assignment due date:** 29<sup>th</sup> of March 2017, 23:59

**Format of the delivery:** PDF

**Method of delivery:** Submit through [Devilry](#) within the deadline.

**Note:** The assignment consists of three parts. It should be solved in groups of 2 – 4 students, unless you have been granted other arrangements.

Be precise when answering the questions. The answers should not be too short or too long. You may write in Norwegian.

If you find aspects of this assignment confusing or ambiguous, do not hesitate to contact any of the group teachers for guidance.

# Part 1

Answer the following questions in your own words. You are not permitted to copy anything from other resources.

## Exercise 1

Explain, in your own words, the seven testing principles. Use examples to show their purpose. Do you have objections to any of the principles?

## Exercise 2

- a) Use the V-model to explain how testing can be integrated in the life cycle of a software development project. What are the benefits of using the V-model with regards to the quality of the system, as well as the time and resources used in the development process?
- b) Is it possible to use the V-model in agile programming development projects? If so, explain how.
- c) Explain to what extent the V-model ensures the seven test principles from Exercise 1 are safeguarded.
- d) What is the different between a test level and a test type?  
Explain the four different:
  - Test levels
  - Test types

## Exercise 3

- a) Explain, in your own words, the difference between static and dynamic testing techniques.
- b) Explain the different steps in a formal review.
- c) A “*walkthrough*” and an “*inspection*” are two different types of reviews. Explain the difference between them; including their purpose and the roles of the participants.

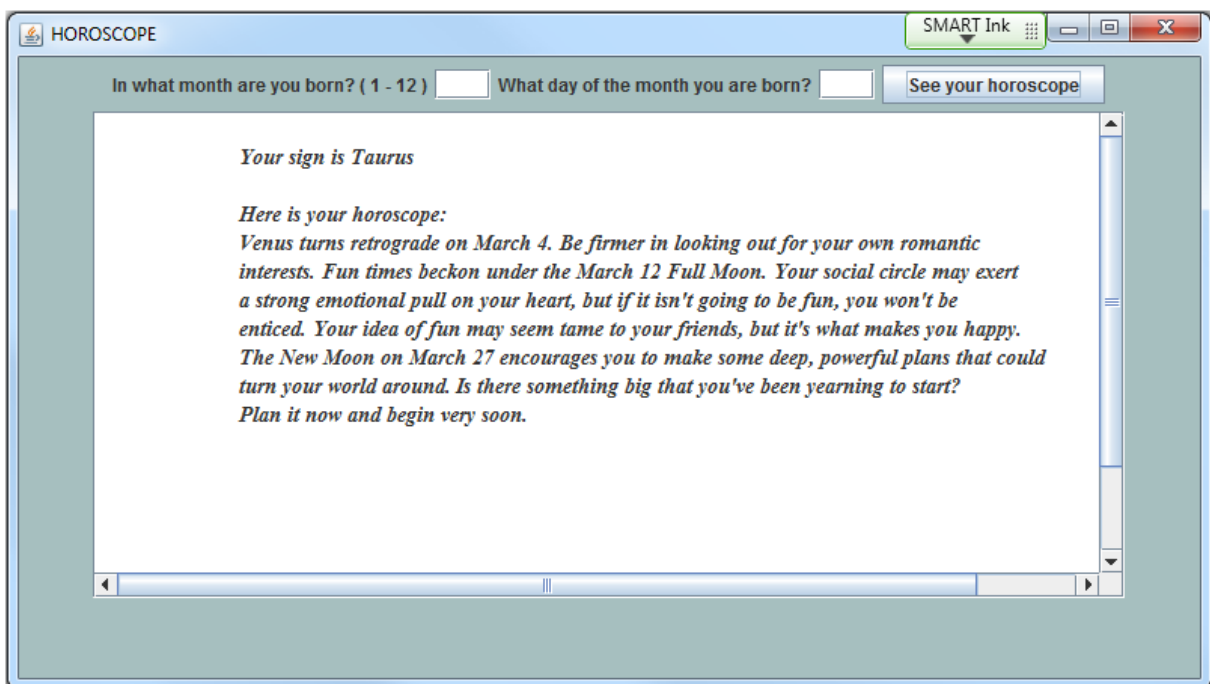
## Exercise 4

- a) Why are specification-based testing techniques also called “Black-box” techniques?
- b) Why are structure-based testing techniques also called “White-box” techniques?  
What is usually the purpose of these techniques?
- c) When do we use experience-based specification-based techniques? What are the benefits and drawbacks related to these techniques?
- d) Would you say that some testing techniques are better than others? What is your opinion on the use of the different techniques?


## Part 2

### Exercise 5

In this exercise you shall use *equivalence partitions* and *boundary value analysis* to detect failures in a little Java program that sets your horoscope based on your birthday. See the user interface below:



You will see the various dates for the different zodiac signs below:

 <b>Aries</b> MARCH 21 - APRIL 19	 <b>Taurus</b> APRIL 20 - MAY 20	 <b>Gemini</b> MAY 21 - JUNE 20	 <b>Cancer</b> JUNE 21 - JULY 22
 <b>Leo</b> JULY 23 - AUGUST 22	 <b>Virgo</b> AUGUST 23 - SEPTEMBER 22	 <b>Libra</b> SEPTEMBER 23 - OCTOBER 22	 <b>Scorpio</b> OCTOBER 23 - NOVEMBER 21
 <b>Sagittarius</b> NOVEMBER 22 - DECEMBER 21	 <b>Capricorn</b> DECEMBER 22 - JANUARY 19	 <b>Aquarius</b> JANUARY 20 - FEBRUARY 18	 <b>Pisces</b> FEBRUARY 19 - MARCH 20

**a)** Determine the different valid equivalence partitions.

Can you determine some invalid partitions?

**b)** Determine the boundary values.

You shall now test the program by using the boundary values and values from the equivalence partitions. (Click [here](http://www.cs.hioa.no/~evav/hioa-dev/YourHoroscope.jar)<sup>1</sup> and choose 'open' to run the program.)

**c)** There are at least three defects / incidents in this program. Can you find them?

Consider the degree of severity of the failures. In what order will you prioritise to correct the defects?

**d)** Use the template below to write an incident report about the defects / incidents you have discovered.

---

<sup>1</sup> <http://www.cs.hioa.no/~evav/hioa-dev/YourHoroscope.jar>

**IEEE 829 STANDARD:  
TEST INCIDENT REPORT TEMPLATE**

Test incident report identifier

Summary

Incident description (inputs, expected  
results, actual results, anomalies,  
date and time, procedure step,  
environment, attempts to repeat,  
testers and observers)

Impact

## Exercise 6

The Oslo City Council wants to introduce a new system for calculating road charges for passage of the toll ring around Oslo. You shall now use decision tables to determine all combinations of conditions to explore the business rules that should be tested according to this system. In addition to introducing a new rush hour charge, the new prices aim to reduce the amount of air pollution in the city. In this exercise, we have specifically chosen to disregard buses and trailers.

- The “rush hour” is defined as the period from 07:30 until 09:30, and later from 15:30 until 17:30.
  - Electric vehicles are not charged outside the “rush hour” period, but are charged a fee of 30 NOK during peak times.
  - Petrol- and diesel vehicles are required to pay 50 NOK outside the “rush hour” period, and get an additional 100 % increase in the price during peak times.
- 
- a) Set up a decision table with the different business rules, including the conditions and related price. There should be one rule for each combination of conditions.
  - b) If there are any combinations (rules) that should *not* occur, you can mark these combinations (rules) with an ‘X’ in the action part of the table. Reduce the number of rules by simplifying (rationalising) the decision table, without losing any of the system functionality.

## Part 3

**Scenario:** A check-in machine is installed at an airport. A detailed description of how to use this machine is as follows:

*To use the check-in machine adhere to the following procedure:*

1. *You can choose to check in with your booking number or with your credit card:*
  - a. *You have chosen to check in with your booking number. Enter your number and click “Continue”.*
    - i. *The machine validates your booking numbers*
    - ii. *If your number is correct, the machine goes to 2*
    - iii. *If the number is not correct, the machine goes back to 1.a., where you are able to try again*
  - b. *You have chosen to check in with your credit card. Put your card into the machine.*
    - i. *The machine validates the card*
    - ii. *If the card is accepted, the machine goes to 2*
    - iii. *If the card is rejected, the machine goes to 1*
2. *You will now be prompted for the number of luggage items you want to check in. When you have entered the number, you click “Continue” and the machine goes to 3*
3. *You will now be asked if you want to reserve a seat or change an existing seat reservation (if you have already reserved a seat)*
  - a. *If you confirm, the machine goes to 4*
  - b. *If you do not confirm, the machine goes to 5*
4. *You choose an available seat and click “Continue”. The machine goes to 5.*
5. *Your boarding card will be printed out along with the luggage tags. When you have taken the boarding card and luggage tags, the machine goes to 6.*
6. *The machine wishes you an enjoyable flight, and goes back to 1.*

*During interaction with the machine, you always have the possibility to click “Cancel”. If you do so, the machine will immediately go back to 1.*

## Exercise 7

In this exercise you shall use *State Transition diagrams* and *State Transition tables* to describe the use of the check-in machine.

- a) Draw a *state transition diagram* that shows how the machine works
- b) Set up a *state transition table* showing every legal *and* illegal (impossible) state
- c) How do you measure
  - i. Statement coverage?
  - ii. Transition coverage?
- d) Write down a sequence of states to achieve 100 % transition coverage. What is the state coverage of the same sequence? What is the relationship between transition coverage and state coverage?

## Exercise 8

In this exercise, you shall *use case testing* to test the check-in machine.

- a) What kind of scenarios will be the main focus in your use case?
- b) Write a testing procedure to test the check-in machine.