Grading

This exam contains five sections. Each section examines the student with respect to one or two of the learning goals specified in the course plan. In order to pass the exam, the student therefore, needs to pass all five sections. For your convenience, each section list the relevant examination criteria.

==========

Section 1

Examination criteria: (i) The student can describe the process for model-driven test design (mdtd) as well as its different activities and (ii) The student can describe the advantage of agile approaches such as test-driven development (tdd) and briefly describe the tdd approach.

Question 1

MDTD allows one test engineer to do the math and leaves the rest of the activities (finding test values, automation, execution etc.) to traditional testers and programmers. For each of the below artifacts developed in an MDTD process, **describe** and **exemplify** what the test engineer does to create it.

1. Model/structure
2. Test requirements
3. Refined test requirements / test specification

Question 2

In a TDD context, what are user stories and how are they used for acceptance test? Examplify!

The level of detail and examples determine the grade.

==========

Section 2

Examination criterion: The student can explain the limitations of software testing

Question 3

In TDD, testing is put first and a large set of regression tests is built up and run after each software update. Are these tests sufficient to test the software well? Will it find most of the faults and should we feel confident that the software is reliable if all tests passes? Motivate your answer and exemplify.

The motivation and example determine the grade.

==========

Section 3

Examination criterion: The student can explain given test techniques in sufficient detail

Question 4

Correlated active clause coverage (CACC):

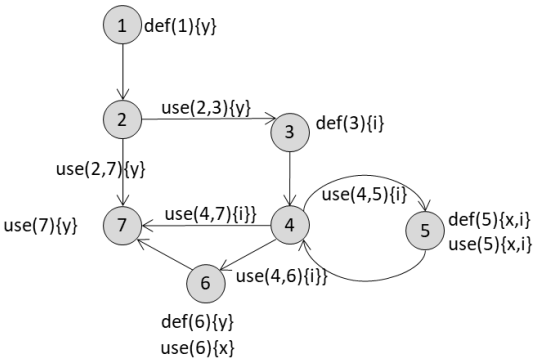
1. Describe the technique (i.e., how it works)
2. List the CACC test requirements for predicate P (i.e., required rows) based on the table below. **Give all feasible alternatives for each clause.**

Row	*A*	*B*	*C*	*Predicate P*	P_A	P_B	P_C
1	T	T	T	T	T	F	T
2	T	T	F	F	T	F	T
3	T	F	T	T	T	F	T
4	T	F	F	F	F	F	T
5	F	T	T	F	T	F	T
6	F	T	F	T	T	T	T
7	F	F	T	F	T	F	F
8	F	F	F	F	F	T	F

Question 5

All-du-path coverage

1. Describe the technique (i.e., how it works)
2. List the du-paths you need for 100% du-path coverage of the below graph



Question 6

Mutation, the ROR mutation operator

1. Describe the ROR operator (i.e., how it works)
2. List the ROR mutants for the given code fragment

```
int count = 0;
for(i=0; i < x.length; i++){
   if (x[i]%2 == 1 || x[i] > 0) {
      count++;
   }
}
return count;
```

Question 7

For a higher grade also: Compare and contrast all-defs and all-uses coverage. Your comparison should focus on (*i*) the effectiveness with respect to the probability to expose failures, (*ii*) the cost with respect to number of test requirements, and (*iii*) the usability with respect to infeasible test requirements.

==========

Section 4

Examination criteria: (i) The student can explain the common concepts in software testing and test automation and (ii) The student can describe the main functionality given by a test automation framework such as JUnit.

Question 8

Four requirements have to be satisfied by a test in order for a test engineer to observe a failure when running it. Two of these requirements are *infection* and *propagation*. Describe what these two requirements mean.

Question 9

What are *test requirements* and *test criteria*? Explain the two concepts and **exemplify**.

Question 10

One of the features in JUnit is fixtures with two types of methods, *@Before* and *@After*. Give an example of each and describe what they do. Make sure that the description is detailed enough to explain when the methods are executed.

==========

Section 5

Examination criterion: The student can give at least one very good argument for why the use of coverage criteria help testers get high-quality tests.

Question 11

In this course we have discussed several advantages of using coverage criteria. One of the advantages concerns easier maintenance of the test set. How does the use of coverage criteria support maintenance decisions such as which tests to remove from, update or add to the test set?

Number of arguments and level of detail determines the grade.