The exam is 75 minutes long. At least 40% of the points is required from both parts separately to pass the exam.

Theoretical part (___ / 25)

Multiple choice questions Select the correct answer(s). There can be several correct choices in each question. There are no penalties for wrong answers, but only fully correct answers get points.

1. (1 point) Which verification technique can be used to check requirement analysis?
   ☐ checklists   ☐ static analysis   ☐ simulation   ☐ reviews

2. (1 point) Which static analysis method produces no false alarms?
   ☐ pattern matching   ☐ abstract interpretation   ☐ model checking   ☐ theorem proving

3. (1 point) Which of the following verdicts signal that there is a problem with the system under test?
   ☐ pass   ☐ fail   ☐ error   ☐ inconclusive

4. (1 point) What are possible goals of software testing?
   ☐ support release decisions   ☐ detecting defects   ☐ finding the cause of a failure   ☐ preventing defects

5. (1 point) Which of the followings can be used as a specified test oracle?
   ☐ FSM model   ☐ runtime exceptions   ☐ previous versions of the system   ☐ predefined expected values

6. (1 point) What are typical elements in a test plan?
   ☐ test design techniques to use   ☐ product risks   ☐ test execution logs   ☐ test objects

7. (1 point) Which of the followings is not a specification-based test design technique
   ☐ equivalence partitions   ☐ condition coverage   ☐ regression test selection   ☐ pair-wise testing

8. (1 point) A decision consists of one or more
   ☐ statements   ☐ decisions   ☐ conditions   ☐ paths

9. (1 point) Generating tests only from source code can uncover
   ☐ differences w.r.t previous versions   ☐ missing requirements   ☐ runtime exceptions   ☐ algorithmic errors

10. (1 point) Which regression testing technique does not reduce the number of tests to execute?
    ☐ Regression Test Selection   ☐ Test Suite Minimization   ☐ Test Case Prioritization   ☐ none of the others

11. (1 point) How can the asymptotic availability of a system with two non-redundant components be calculated if both components have k_c as their component-level asymptotic availability.
    ☐ $1 - (1 - k_c)^2$ ☐ k_c^2 ☐ $2k_c$ ☐ none of the others

12. (1 point) What kind of architectural elements can be described in the AADL language?
    ☐ devices   ☐ threads   ☐ processors   ☐ mapping between SW and HW

13. (1 point) What are typical parameters for a performance analysis model?
    ☐ scheduling policy   ☐ execution time   ☐ repair rate   ☐ none of the others

14. (1 point) What kind of faults can be detected only by runtime verification and not by design-time techniques?
    ☐ hardware faults   ☐ coding faults   ☐ missing requirements   ☐ none of the others

15. (1 point) Using which of the below items can properties be formalized for runtime verification?
    ☐ temporal logics   ☐ regular expressions   ☐ loop invariants   ☐ design-by-contract assertions

Short answer questions Write precise answers to the following questions inside the boxes.

16. (2 points) Explain what kind of false positive can be produced if a static analysis method is not path sensitive.

17. (2 points) Explain using an example why 100% condition coverage does not imply 100% decision coverage.

18. (2 points) Explain how dynamic symbolic execution extends static symbolic execution.

19. (2 points) In model-based testing what is the difference between online and offline test generation?

20. (2 points) Explain what is the difference between availability and reliability.

Practical part

Question:	21	22	23	24	Total
Points:	5	5	5	5	20
Score:					

21. **Static analysis**

    The following code snippet is given, where the function `ioread32()` reads a 32 bit signed integer.

    ```
        int64 a, b
    1: a = ioread32()
    2: if (a < 0) {
    3:     b = -a
    4: } else {
    5:     b = a
    6: }
    ```

    (a) (3 points) Using abstract interpretation, describe the values of the program variables in each line $(X_1, X_2, \ldots, X_6)$! If no concrete value is known, compute symbolically (using previous values)!

    (b) (2 points) Using abstract interpretation with intervals, propagate the range of the program variables from the entry of the program (X_1)! What is the possible range of variable b at line 6 (X_6)?

22. **Combinatorial testing**

    Our application is available for the iOS, Android and Windows platforms. For Windows it can run also on PC, for the other two only smart phone and tablet is supported. The application is localized to Hungarian and English.

    (a) (1 point) How many different configurations can be tested for the application _____

    (b) (4 points) We do not need all the possible combinations, pair-wise coverage is enough. Give a set of configurations that is minimal and satisfies the pair-wise criterion.

#	Platform	Devise	Language

23. **Model-based test generation**

Given the following finite state machine:
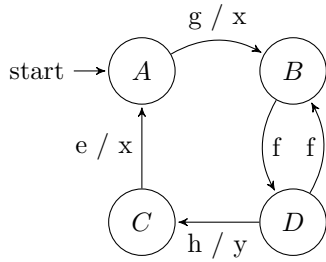
(a) (5 points) Give a test suite that reaches 100% transition coverage!

24. **Structure-based testing**

Given the following source code fragment:

```
int function13(int a, bool b){
    int c = 10;
    if (b) c = −c;

    for (int i = 0; i < a; i++){
        c++;
    }

    return c;
}
```

(a) (3 points) Draw the control flow graph (CFG) of the function next to the code.

(b) (2 points) Give a test suite that reaches 100% decision coverage.