Student name and number: _____

Instructions, Information, and Rules

Please, read these instructions carefully. Failure to comply with *any* of the following instructions means *invalidation of your exam*:

1. You have 1.5 hours to complete the exam.

2. The exam consists of 18 multiple-choice and 6 open questions. For each multiple-choice question, *only one* answer is correct. The questions are shuffled among the attendees.

3. A correct answer to a multiple-choice question will add one point to your grade. An incorrect answer will *decrease* your grade of one point. Not answering a question is *not* considered an incorrect answer.

4. A correct and detailed answer to an open-question will add two points to your grade; A correct but partial answer will add one point to your grade.

5. In principle, given (i) M correct answers and I errors to the multiple-choice questions, and (ii) O correct answers and P partial answers to the open questions, the grade G of the exam will be determined by the formula:

$$G = (((1 * M) - (1 * I)) + ((2 * O) - (1 * P))).$$

The grade G ranges between 0 and 30. It will be increased by up to the three points of the hands-on assignment *if and only if* (1) G will be higher than 18 and (2) the assignment has been delivered by the fixed deadline. Moreover, in case you are interested in further increasing your grade, you can do the *oral* part of the exam this afternoon; however, *this is possible if and only if G is higher than 23*. The outcome of the exam will be made available on the GITHUB repository of the course[1] within three hours since the end of the exam. To book a slot for the oral exam, you must send me an e-mail by this afternoon at 15:00. The oral exam will take place in the SESA LAB at 16:00 and comprise three questions spanning across all topics treated in the course.

6. Fill in your answers on the provided answer sheet and hand it in at the end of the exam.

7. The use of books, papers, computers, phones, smart-watches or other material is *not* permitted.

8. You are *not allowed* to talk to your colleagues; this is forbidden for *both* questions related to the exam and any other form of communication (requests for pens, papers, etc.). All you need for passing the exam is this document and a pen: if you do not have a pen, ask *before* the beginning of the exam. Note that there exist *different forms of communications*: trying to spy the exam of one of your colleagues is considered a forbidden communication.

Good luck!

---

[1]Link here: https://github.com/fpalomba/SWDependability-Unisa2019.

Page left empty on purpose.

Part I
Dependability Requirements

1. Fill in the blank: Some of the methodologies stimulating software reliability are _____.

   A. adoption of most of the DevOps practices and team composition management.

   B. application of data redundancy and management strategies.

   C. testing, code review, and source code quality management.

   D. continuous integration and deployment.

2. Fill in the blank: The maintainability of a software system can be stimulated through _____.

   A. the hiring of database experts.

   B. a definition of strategies to deal with vulnerabilities.

   C. code smell detection and refactoring.

   D. data redundancy and fault recovery approaches.

3. Fill in the blank: The _____ of a software system comprises methods and techniques that allows end-users to always have access to its functionalities.

   A. maintainability

   B. reliability

   C. availability

   D. None of the others

4. Define the dependability requirements and provide an example showing whether and how they are connected to each other. NB: Do not exceed the available space.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Part II

Code Smells and Refactoring

5. Fill in the blank: One of the principles driving refactoring is coupling; this indicates _____.

   A. the degree to which a certain class or module relies on other classes or modules.

   B. how focused the responsibilities of a certain class or module of a system are.

   C. the extent to which a class or module is feasibly testable.

   D. none of the others.

6. Which of the following sentences about *coupling relations* is true?

   A. Structural coupling information is easy to extract from source code, but it is not always available.

   B. Semantic coupling information is not meaningful for developers and it is not always available.

   C. Dynamic coupling information is not meaningful for developers, but it is easy to extract.

   D. None of the above.

7. Consider the case in which you have a class that delegates most of its functionalities to external classes. Fill in the blank: This class is affected by a _____ code smell.

   A. Long Method

   B. Middle Man

   C. Feature Envy

   D. Inappropriate Intimacy

8. Describe the main characteristics making a package affected by the *Promiscuous Package* code smell. Describe also what is the corresponding refactoring operation and what are the steps required to perform it. NB: Do not exceed the available space.

Software Vulnerabilities

9. Select which of the following sentences best relates to *vulnerabilities*:
    - A. All faults introduced by programmers lead to vulnerabilities.
    - B. Code smells are the main cause of software vulnerabilities.
    - C. Vulnerabilities can eventually be treated by external users to access the data storage layer of an application only.
    - **D. None of the above.**

10. Fill in the blank: One of the approaches to identify vulnerabilities is represented by logistic techniques; they consist of _____.
    - A. applying stochastic models to find vulnerable areas of source code
    - **B. running machine learning models trained on previous available vulnerability data**
    - C. analyzing the source code without compiling it
    - D. testing how the system accepts input data and how it manages corner cases

11. Fill in the blank: A typical life of a software vulnerability comprises the _____ cycle.
    - A. Patching-Exploitation-Discovery-Disclosure
    - B. Disclosure-Discovery-Exploitation-Patching
    - C. Exploitation-Patching-Discovery-Disclosure
    - **D. Discovery-Disclosure-Exploitation-Patching**

12. Describe the fuzz testing strategy for detecting vulnerabilities in source code. If possible, also provide an example of fuzz testing-based vulnerability discovery. NB: Do not exceed the available space.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Part IV

Machine Learning for Software Dependability and Defect Prediction

13. Which of the following sentences about unsupervised learning techniques is *true*?

    A. This type of learning is used when the ground-truth data does not need to be labeled.

    B. This type of learning is primarily used when the data needs to be repaired as a consequence of a failure.

    C. The ground-truth definition is key to correctly train an unsupervised machine learner.

    D. None of the above.

14. Fill in the blank: Data imputation techniques _____.

    A. are used to discard highly correlated independent variables

    B. are used in case the training data is complete

    C. are not limited to statistical analysis

    D. none of the above

15. Select the description that best explains how a *supervised learning approach for defect prediction* works:

    A. It aims at predicting the portions of source code that are more likely to be defective in the future by taking into account a set of metrics computed on the classes of a system.

    B. Starting from an unlabeled set of data, it analyzes the structure of source code to identify potential defects.

    C. It aims at predicting which classes will be defective in future releases of a system by only considering the product metric values computed on its current snapshot.

    D. Exploiting process metrics, it learns a classifier that identifies defects in a specific package of the considered project.

16. Describe when the problem of multi-collinearity can appear and a possible solution to deal with it. NB: Do not exceed the available space.

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

Part V
Software Metrics and Traceability

17. Traceability links can be kept using:
    - A. a traceability schedule.
    - B. a traceability timeline.
    - **C. a traceability matrix.**
    - D. a traceability graph.

18. Knowledge-based traceability links can be retrieved:
    - A. by parsing the static structure of the source code.
    - **B. by developers who know the semantics of the source code.**
    - C. by running specific test cases to expose the dynamic behavior of the code.
    - D. by analyzing sequence diagrams.

19. When recovering traceability links, heuristic-based approaches aim at:
    - A. detecting structural relationships between source and test code.
    - **B. identifying knowledge-based links between the implemented artifacts.**
    - C. identifying traceability links only between requirements.
    - D. detecting relationships between UML diagrams.

20. Define the goal of cohesion, coupling, and complexity metrics. Provide also an example metric for each category; finally, focus on the Lack of Cohesion of Methods (LCOM) metric, reporting the steps required to compute it. NB: Do not exceed the available space.

DevOps and Infrastructure Code

21. An *application deployment topology* is:

    A. a graph representing how the procurement of deployment resources works.

    B. is a representation of the teardown strategies of a typical DevOps infrastructure.

    C. a graph of physical artefacts that need support for several lifecycle phases.

    D. None of the above.

22. Fill in the blank: Continuous Integration is a DevOps tactic that _____.

    A. aims at merging all developers' working copies to a shared mainline several times a day.

    B. aims at merging all developers' working copies to a shared mainline once per day.

    C. aims at merging all developers' working copies to a shared mainline once per week.

    D. aims at merging all developers' working copies to a shared mainline whenever a new change request is available.

23. Fill in the blank: Micro-services are a software development practice that _____.

    A. structures an application as a collection of highly coupled services.

    B. structures an application as a collection of services that do not communicate with each other.

    C. structures an application as a collection of loosely coupled services.

    D. does not respect the standard principles of cohesion and coupling of software architectures and classes.

24. Define what is containerization. Also, summarize the difference between Docker and Virtual Machines. NB: Do not exceed the available space.

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____