

# **Final Exam - Study Guide**

## **Time/Location:**

Dec 11<sup>th</sup> Thursday from 2.30 - 4.20 pm

The duration of the exam is 1 hour and 15 minutes.

**Location :** in-class , on Canvas using Lockdown Browser.

The setup is same as exam 2. You must be physically present in the classroom to take the exam.

---

**Note: You can bring a cheat sheet 4 pages written or typed (front and back). Please note that you are required to submit your cheat sheet after taking the final exam when you sign out.**

## **Approximate point distribution from each topic (approximately):**

|                                                                               |     |
|-------------------------------------------------------------------------------|-----|
| Software Quality Assurance (Module 1)                                         | 5%  |
| Introduction to software testing (Module 2 and 3)                             | 10% |
| Functional Testing (Modules 3, 4 and 5)                                       | 15% |
| Bug Management (Module 6)                                                     | 5%  |
| Introduction to Code based (Structural Testing) –Module 7, 8 and Junit/TestNg | 15% |
| Testing object-oriented applications and Integration testing – Module 13      | 20% |
| Testing web applications (Module 11 and 12)                                   | 20% |
| Static Testing/Reviews (Module 9 and 10)                                      | 15% |

**Question Types:** The final exam will have multiple-choice/true-false questions and short answer questions. The exam will be auto graded.

## **Topics**

### **1. Software Quality Assurance**

- What is software quality
- Quality Assurance Techniques
- Role of software testing in quality assurance

## **2. Introduction to software testing**

- Venn diagram view of software testing
- , Functional vs. Structural Testing
- Black box vs. White box testing

## **3. Functional Testing**

- Equivalent classes
- Boundary value analysis
- Decision Table
- Model based testing
- Strengths and weaknesses of above four techniques

## **4. Bug Management**

- Anatomy of a bug report
- The bug workflow
- The Weibull Curve

## **5. Code based testing**

- What is structural testing - model programs as a graph(s)
- Basis graph terminology
- Indegree, outdegree
- sink, source, and transfer node
- Cyclomatic Number
- Control path testing and its limitations
- Data path testing (D-U path)

## **6. Testing object-oriented applications and Integration testing**

- Design considerations of OO applications
  - C-K matrix
  - Coupling and Cohesion
  - Type of objects and their role in the design (control, boundary, entity classes)
  - Refactoring
  - MVC architecture principles
- Code smells and refactoring
- StateCharts
  - Developing stateCharts and designing test cases based on state charts
- Designing test cases based on use cases and sequence diagrams
- Integration testing
  - Top down, bottom –up, and sandwich approach
  - Integration testing for OO applications (using sequence diagrams, stateCharts, use-cases, sequence diagrams, and collaboration diagrams to design test cases)

## **7. Testing web applications**

- What is distributed software (conventional three-tier client-server, service-oriented, cloud architectures)
- What is gray-box testing
- Why gray-box approach is better in testing web applications
- Challenges in Web application testing
- Quality dimensions of web applications, how web application design and test design of web applications related to each other

- Approaches in testing **three tier client-server and SOA applications** (how the software architecture of such applications influences in designing test cases)
- Petri-net- advantages of Petri-net
- Using Petri-net to model web service workflow applications

**DO THE SAMPLE FINAL EXAM**