

**Computer Science Department**

**Software Testing and Quality Assurance**

**COMP438**

**COURSE OUTLINE**

**Spring 2023**

**Instructors**

Dr. Majed Ayyad

**Course Description**

The goal of the course is to provide students with the skill to select and apply testing techniques and principles that are appropriate to a particular system (database, web, mobile) and appropriate to the stage in the development life-cycle (requirements analysis, design, coding and release). Furthermore, students will acquire the knowledge on how to select the appropriate test strategy (e.g., Unit testing, Black box testing, White box testing, integration testing, profiling, test driven development, acceptance testing) and tools that fit the problem under consideration. The course will cover modern aspects of testing techniques that is aligned with Agile and DevOps approaches as well on learning the importance of automated testing for fast release and deployment.

### يهدف هذا المساق الى تزويد الطلبة بالمهارات والمعرفة اللازمة لاختيار وتطبيق التقنيات والمبادئ الأساسية لاختبار البرمجيات وفحصها وضمان جودتها. من خلال هذا المساق سيتعلم الطلبة كيفية اختيار تقنيات الفحص حسب نوع النظام (قواعد البيانات، أنظمة الوب وأنظمة الموبايل) كما سيتم دراسة الطرق المختلقة لاختبار البرمجيات وفقا لمرحلة التطوير (متطلبات النظام، مرحلة التصميم، مرحلة البرمجة، ومرحلة الاطلاق). بالإضافة الى ذلك سيتعلم الطلبة الاستراتيجيات المختلفة لعمل الاختبارات مثل (اختبارات القبول والتصميم واختبارات وحدات البرمجة واختبارات التكامل واختبارات التراجع والصندوق الشفاف والصندوق المغلق وغيرها). سيغطي هذا المساق العديد من المواضيع التي تتلاءم مع التطورات الحديثة في البرمجة مثل الحاجة الى معرقة التقنيات التي تساعد في أتمتمة عمليات الفحص وتسريعها والتي تتناسب مع فلسفة التطوير المرن وعمليات الدمج بين التطوير والتشغيل للبرمجيات.

**Objectives**

On completion of this course, the student will be able to:

* Relate software testing to all software development processes
* Understand how testing works with new paradigms such as DevOps and Continuous Integration, Delivery and Deployment and automated testing.
* Develop test strategies based on requirements analysis
* Design and implement comprehensive test plans with instrumented code
* Apply different testing techniques
* Compute test coverage and yield
* Understand the limitation of different testing process
* Gain knowledge on how to implement automated testing

**Outline**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Topic** | **Material** | **# of Lectures (75 min)** |
|  | Introduction to software testing and theoretical background   * Testing mindset * V-Model * Coverage | [2] Chapter 1  Lectures notes | **3** |
|  | Overview:   * Structural Coverage Criteria * Graph Coverage for Source Code | Lecture notes | **2** |
|  | Testing Process   * Test Case Writing * Test Cycles * Generating Test Reports * Testing reports | [2] chapter 5  Lecture notes | **3** |
|  | Black-Box Testing   * Equivalence Partitioning * Boundary-Value Analysis * Decision Table Testing * State-Transition Testing * Lab | [1] chapter 4-section-I | **5** |
|  | White Box Testing   * Statement Coverage * Decision Coverage * Condition Coverage * Path Coverage * Loop Coverage * Lab | [1] chapter 4 -section-11  ; lecture notes | **5** |
|  | Codeless test Automation - - Using Selenium IDE | | Lecture notes | **3** |
|  | Module (Unit) Testing  Top-Down versus Bottom-Up Testing | [1] chapter-5 | **2** |
|  | Performance Testing using JMeter  Response time, Throughput, Utilization, & Robustness | Lecture notes | **3** |
|  | Usability (User) Testing | [1]] chapter-7 | **2** |
|  | Project presentation |  | **2** |
|  | **Total** |  | **30** |

**Grading Criteria**

The course will include several evaluations including assignments, exams/quizzes and a main project. The grading distribution is subject to change depending on university and computer science department regulations during emergency situations.

Midterm 30%

Assignments and/or Quizzes 10%  
Project 20%

Final Exam 40%

**Pre-requisites**

Students should have some basic understanding of software engineering and software development lifecycle together with experience of programming preferably Object-Oriented Programming and basic knowledge of data structure knowledge

**Textbook**

1. [The Art of Software Testing ,](https://www.amazon.com/Art-Software-Testing-Glenford-Myers/dp/1118031962) [[Glenford J. Myers](https://www.amazon.com/Art-Software-Testing-Glenford-Myers/dp/1118031962)](https://www.wiley.com/en-us/search?pq=%7Crelevance%7Cauthor%3AGlenford+J.+Myers)[,](https://www.amazon.com/Art-Software-Testing-Glenford-Myers/dp/1118031962)[[Corey Sandler](https://www.amazon.com/Art-Software-Testing-Glenford-Myers/dp/1118031962)](https://www.wiley.com/en-us/search?pq=%7Crelevance%7Cauthor%3ACorey+Sandler)[,](https://www.amazon.com/Art-Software-Testing-Glenford-Myers/dp/1118031962)[[Tom Badgett](https://www.amazon.com/Art-Software-Testing-Glenford-Myers/dp/1118031962)](https://www.wiley.com/en-us/search?pq=%7Crelevance%7Cauthor%3ATom+Badgett) [ISBN: 978-1-119-20248-6 September 2015](https://www.amazon.com/Art-Software-Testing-Glenford-Myers/dp/1118031962)
2. Software Testing and Analysis, by Mauro Pezzè and Michal Young, John Wiley & Sons, 2008

**References**

1. Software Testing Foundations: A Study Guide for the Certified Tester Exam by Andreas Spillner, Tilo Linz, Hans Schaefer. Note students may use this as an example for the course project.
2. AGILE TESTING: A PRACTICAL GUIDE FOR TESTERS AND AGILE TEAMS by Lisa Crispin Janet Gregory