Department of Computer and Information Sciences
COSC 442 – Software Quality Assurance and Testing

Spring 2023

Instructor:	Dr. Josh Dehlinger	**Email:**	jdehlinger@towson.edu
Office:	7800 York Road, Room 476	**Slack:**	https://cosc442dehlinger.slack.com
Office Hours:	MW 11:00-12:00 By email and appointment	**Phone:**	410.704.4536 (Slack/email preferred)

Course Hours:	MW 9:30 – 10:45
Course Location:	YR 304
Course Website:	Blackboard

Required Textbook: Foundations of Software Testing, 2nd edition
Aditya P. Mathur Pearson, 2014
ISBN-13: 978-8131794760
ISBN-10: 8131794768

Prerequisite: COSC 412 – Software Engineering

Course Description: Software is becoming ubiquitous in society, steadily increasing the demand for more reliable and secure software. As a result, software testing and quality assurance measures now account for more than half of the cost of software development and developers spend 50% of their time performing software testing activities. Further, emerging technology areas (e.g., web applications, mobile applications, cyber-physical systems, secure systems, etc.) require software that is tested more thoroughly. This course will cover the technical and practical methods that software engineers and developers can use to test and evaluate software quality assurance and equip students with the software quality assurance and testing concepts, techniques and tools to develop high-quality software.

Course Objectives: Upon completion of the course, students will understand and be familiar with software testing concepts, techniques and tools for the testing, maintenance and evolution of software systems. Specifically, students will be able to:

- Understand and communicate basic software testing terminology, principles and techniques
- Know the testing process, black-box and white-box testing techniques used in software testing
- Develop unit, module, subsystem, integration and system test cases for a variety of systems including object-oriented, web-based, concurrent, embedded and distributed systems
- Apply quantitative, technical and practical testing methods and tools for testing and modifying evolving software

Course Topics: This course develops and details the principles, techniques, tools and practices of software testing. The course will survey current software testing techniques and tools in combination with application homework assignments to provide students with practical experience. Potential course topics, in the order they may be presented, could include:

- Test generation from requirements
    - Equivalence partitioning
    - Boundary-value analysis
    - Category-partition method
- Test generation from finite state models

- Test generation from combinatorial designs
- Test adequacy
    - Control-flow analysis
    - Data-flow analysis
    - Program mutation
- Unit testing
- Integration testing
- Regression testing
- Fuzz testing
- System and acceptance testing
- Testing different technologies
    - Testing for web-based systems and web services
    - Testing graphical user interfaces
    - Testing mobile applications
- Testing/auditing tools
    - Static and dynamic analysis tools

Bibliography/References: References and possible assigned readings may include some of the following:

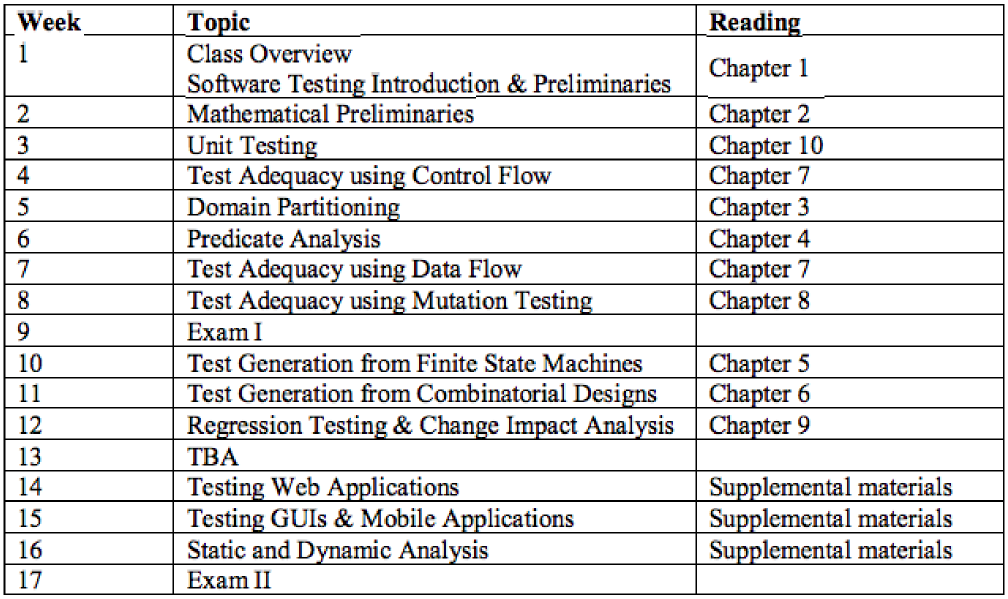
- *Software Engineering Body of Knowledge*, Version 3, Chapter 5, 2022.
- *Software Testing – A Craftsman's Approach*, 3rd edition, Paul C. Jorgensen, 2008
- *Software Testing, The Computer Science and Engineering Handbook*, Gregory Kapfhammer, CRC Press, http://cs.allegheny.edu/%7Egkapfham/research/publish/software_testing_chapter.pdf
- *Why Programs Fail: A Guide to Systematic Debugging*, Andreas Zeller, Morgan Kaufmann, 2005

Course Format: Active learning techniques, lectures, discussion sessions (in class AND online via Slack), presentations, lab exercises and projects may be used. ***Students are expected to read the textbook*** and find and use current content on the course subjects using the library, internet and provided resources. Some assignments will necessitate collaborative learning while others will require individual research and presentation.

Attendance Policy: Students are expected to attend all classes, or watch recorded lectures, to remain current in the coursework. It is the student's responsibility to remain current on the handouts, assignments and notes if class is missed. The instructor will allow only students with documented excuses (see below) to make up missed work or assignments *when it is feasible*.

If the student is absent from an exam during the scheduled time for that exam, the student will automatically receive a grade of zero (0) for the exam unless: (a) the student notifies the instructor of the absence prior to the exam; (b) the student is ill and supplies a written doctor's excuse explaining the absence; or (c) there is an extraordinary situation which the instructor allows as an acceptable excuse. Only under one of these circumstances will arrangements for a makeup exam will be made.

Tentative Schedule: The following is a tentative schedule. Note that these topics and chapters are subject to change based on time and discretion of the instructor and interest of students. An updated schedule will be maintained and posted on Blackboard.



Week	Topic	Reading
1	Class Overview Software Testing Introduction & Preliminaries	Chapter 1
2	Mathematical Preliminaries	Chapter 2
3	Unit Testing	Chapter 10
4	Test Adequacy using Control Flow	Chapter 7
5	Domain Partitioning	Chapter 3
6	Predicate Analysis	Chapter 4
7	Test Adequacy using Data Flow	Chapter 7
8	Test Adequacy using Mutation Testing	Chapter 8
9	Exam I	
10	Test Generation from Finite State Machines	Chapter 5
11	Test Generation from Combinatorial Designs	Chapter 6
12	Regression Testing & Change Impact Analysis	Chapter 9
13	TBA	
14	Testing Web Applications	Supplemental materials
15	Testing GUIs & Mobile Applications	Supplemental materials
16	Static and Dynamic Analysis	Supplemental materials
17	Exam II	

Grading Policy: Students will be evaluated on the following basis:

Homework/Lab Projects: 40%
Final Project: 20%
Quizzes: 10%
Exams: 30%

Final course grades will abide by the following scale:

A	95-100	**B-**	80-82.9	**D**	63-66.9
A-	90-94.9	**C+**	75-79.9	**D-**	60-62.9
B+	87-89.9	**C**	70-74.9	**F**	Below 60
B	83-86.9	**D+**	67-69.9		

Assignments/Projects: There will be several lab programming mini projects given throughout the semester consisting of designing/developing test cases (i.e., unit, integration, regression and acceptance), running automated testing suites and evaluating testing tools on various types of software including graphics-user interfaces, web services, mobile applications, object-oriented applications taken from popular, open-source projects. Lab projects are meant to provide a "hands on" experience to students in order to emphasize the material recently covered during lecture. Lab projects are intended to be highly interactive in which students are encouraged to seek help to fully understand the material. All projects will be given to provide students ample time to complete. Students should start early and frequently back up their work and all submissions should be *well documented giving credit to respective sources with proper citations*. Students are reminded that the instructor is available for help during office hours and otherwise through email, Slack or appointments. Homework may also be assigned in the lecture at the instructor's discretion and as the need arises.

Final Project: The final course project allows you to demonstrate that you've achieved the objectives of this course: creating and evaluating test suites, using tools for software maintenance and verification, and working with reasonably-sized software systems. The final course project will be done in small groups and will consist of testing, refactoring, patching and documenting an active, open-source project. Further information on the final project will be provided shortly.

Quizzes: There will be five (5) announced, graded, in-class/online quizzes covering the main ideas from the assigned textbook chapters and lectures.

Exams: There will be 2 exams for this course, weighted as 15% each. The dates of the first exam will be announced during the semester. Make up tests can only be given in extenuating and documented circumstances and must be arranged in advance. The second exam will be given, as scheduled, during exam week on Friday, May 19, 2023 from 8:00 – 10:00AM.

Late Policy: All assigned work (i.e., assignments and projects) is expected to be completed and submitted by the stated deadline. ***No late work will be accepted*** and a grade of zero (0) will be given. Students are encouraged to discuss homework and projects as a means to share knowledge, experience and lessons learned as part of the learning process, but academic honesty should be strictly observed (see below).

Posting of Grades: University policy prohibits posting of grades in any form. The instructor will not report grades via email or in response to phone calls. Grades for the semester can be accessed online via Blackboard.

Cheating and Plagiarism: Academic honesty is strongly observed. This course may consist of both individual and team assignments. A team project is an assignment in which collaboration is allowed and highly encouraged. However, the work of the team *must* be of the team's creation and not plagiarized from other sources. Individual assignments *must* reflect the work of the individual student and of his/her creation. While studying together, discussion and collaboration is encouraged, individual assignments *must be individually prepared* – copying or sharing files, diagrams and/or code is considered cheating. The penalty for cheating will, at a minimum, consist of a grade of zero for the dishonest work and may lead to the possibility of **course failure** depending on the severity. Students are responsible for reading and knowing Towson University's policy regarding academic dishonesty, located at https://www.towson.edu/about/administration/policies/03-01-00-student-academic-integrity-policy.html.

Blackboard Website: There will be a Blackboard website for this course. Students will be responsible for frequently checking the site for updates and announcements. All course related materials will be available for download from the Blackboard site.

Computer Lab Information: Lab hours and policies for the COSC and Cook Library can be found at http://www.towson.edu/fcsm/departments/computerinfosci/resources/labs.html and here.

Classroom and Lab Policy: Food and drink is not allowed in the labs and classrooms with the exception of water in the classroom only. All cell phones should be turned off or put on silent to avoid disruptions and distractions.

Email Policy: All email correspondence with the instructor must be conducted using the student's Towson University email account. The instructor will not read/respond to any email messages from outside accounts.

Repeat Policy: Students may not repeat a course more than once without prior permission of the Academic Standards Committee.

If you have a learning disability and/or need accommodation for any reasons, please advise the instructor as early as possible in the course.