



College of Engineering, Physics, and Computing

CSC 635 – Software Verification, Validation, and Testing (3 Credits) Spring 2026

Course Description

The development of modern software systems, like all engineered products, requires a rigorous and systematic testing phase before they can be considered suitable for deployment. In software engineering, however, testing occupies an especially prominent role, often constituting one of the largest investments of time, labor, and resources within the entire development lifecycle. The scale, heterogeneity, and continual evolution of contemporary software make the existence of defects not merely possible but fundamentally unavoidable. Engineers must therefore employ a spectrum of verification, validation, and testing strategies to identify, isolate, and mitigate these defects within the practical constraints of budget, schedule, and expected system reliability.

Software is ultimately a sequence of logical instructions that governs the temporal evolution of a computing system, transitioning from inputs through complex intermediate states to produce outputs. When faults exist, these instructions may drive the system through unintended states, yielding incorrect or even unsafe behavior. Importantly, the correctness of a software product cannot be guaranteed solely through abstract reasoning or logical consistency. True validation arises only through the interaction between the software and the underlying physical system on which it executes. This interaction is inherently unpredictable due to three persistent challenges: (1) intermediate computational states are only partially observable; (2) the space of possible inputs is effectively unbounded; and (3) operating environments introduce variability and disturbances that are difficult, if not impossible, to model completely.

This course examines the theoretical underpinnings and practical realities of ensuring software quality under these constraints. Students will explore the principles of verification and validation (V&V), the nature of software faults and failures, and the strategies used to detect, prevent, and reason about defects. Emphasis is placed on both static and dynamic testing techniques, test planning, inspections, debugging methods, and the evaluation of software behavior under realistic operational conditions. Case studies drawn from real-world failures and industry practices will illustrate how testing contributes not only to functional correctness but also to reliability, safety, and long-term maintainability.

Prerequisites

N/A

Instructors Information

- Dr. Sepideh Akhbarifar
 - Research Scientist at Vitreous State Laboratory | Adjunct Associate Professor, Department of Mechanical Engineering

- E-mail: 05akhbarifar@cua.edu
 - Office: 312 Hannan Hall
 - Telephone: 2023196156
 - Office Hours: By appointment
- Dr. Richard Kelley
 - Clinical Assistant Professor, Department of Computer Science
 - E-mail: kellyrc@cua.edu
 - Office: 324 Pangborn Hall
 - Telephone: 2023196942
 - Office Hours:

Communication

- Email is the preferred method of communication for all course-related matters.
- In the event of schedule changes or emergencies, students will be notified via Brightspace announcements and email.

Class Meeting

- Monday 5:10 PM -7:40 PM
- Hybrid Format
- Attendance is required for all lectures. Punctuality is expected.
- Zoom information

Topic: CSC 635

Time: Jan 12, 2026, 05:00 PM Eastern Time (US and Canada)

Every week on Mon, until May 11, 2026, 18 occurrence(s)

Join Zoom Meeting:

<https://cua.zoom.us/j/87018702610?pwd=KyduitbgANxVjnlDhz4xzilg3ZeGj.1>

Meeting ID: 870 1870 2610

Passcode: 277470

Join instructions:

https://cua.zoom.us/meetings/87018702610/invitations?signature=JyJ_tfR66J_WxxV5WVD6dUAV81iNulPN56AA6FbJbsM

Instructional Methodology and Course Materials

The contents of this course will be delivered through a combination of online and in-person sessions and will be co-taught by two instructors. Instruction will include conceptual explanations, visual materials, and examples presented through case studies or problem scenarios. Most instructional content will be provided in the form of digital slide presentations, which will be made available to students after each class.

All course materials, along with announcements, reminders, and other important information, will be posted and communicated through Brightspace.

Textbooks

There is no required textbook for this course. However, the following references are recommended for students who would like additional background or deeper study:

- *Software Testing: Principles and Practices* by Naresh Chauhan
- *Foundations of Software Testing* by Ammann & Offutt

- *Software Engineering* (V&V chapters) by Ian Sommerville

Additional articles, examples, and online resources will be provided throughout the semester.

Grading Scheme

Weight

Participation/Attendance (30%); Assignments (20%); Final Exam (50%, take-home)

Scale

A (93-100 %); A- (87-92 %); B+ (80-86 %); B (73-79 %); B- (67-72-82 %); C (60-66 %); F (<60 %)

Lecture & Homework

Course materials will be shared via Brightspace.

Academic Honesty

Academic integrity is expected of all students. Refer to the full policy at: <http://policies.cua.edu/academicundergrad/integrityfull.cfm>.

Course Add/Drop deadline is Friday, January 23.

Tentative Course Schedule

Week	Date	Tentative	Lecture Topics
1	Mon: 01/12	Zoom	Introduction to Software Engineering & Software Testing
2	Mon: 01/19	No Class	Rev. Martin Luther King, Jr., Day: Holiday; No Class
3	Mon: 01/26	Zoom	Foundations: Software Quality, Faults/Errors/Failures
4	Mon: 02/02	Onsite	Verification vs. Validation; V-Model
5	Mon: 02/09	Onsite	Static Testing: Inspections, Reviews, Checklists
6	Tue: 02/10	Zoom	Administrative Monday: Static Analysis Tools (Linting, Control Flow, Data Use)
7	Mon: 02/16	Zoom	Dynamic Testing: Test Levels (Unit, Integration, System, Acceptance)
8	Mon: 02/23	Onsite	Black-Box and White-Box Test Design Techniques
9	Mon: 03/02	Zoom	Test Case Design: Equivalence Partitioning & Boundary Value Analysis
10	Mon: 03/09	No Class	Spring Break: No Class
11	Mon: 03/16	Zoom	Test Planning: Requirements Traceability, Documentation
12	Mon: 03/23	Onsite	Debugging vs Testing: Strategies & Failure Analysis
13	Mon: 03/30	Onsite	Software Testability & Non-Functional Testing (Reliability, Performance)
14	Mon: 04/06	No Class	Easter Monday: No Class
15	Mon: 04/13	Onsite	Advanced Topics: IV&V, Cleanroom, Formal Methods

			(Overview)
16	Mon: 04/20	Zoom	Industry Practices: CI/CD Testing, Automated Testing Frameworks
17	Mon: 04/27	Onsite	Review Session & Applied Case Study Workshop
18	May 5–May 9	<i>Final</i>	<i>Take-Home Exam:</i> Release & due dates will be announced during the Final Exam Period