University at Buffalo
The State University of New York

CSE306 - Software Quality in Practice
Fall 2024

COURSE INFORMATION

Lecture times – T/Th 8:00 – 8:50
Location – Davis 101

Lab times – A1 T/Th 20:00 – 22:00; A2 T/Th 16:00 – 18:00; A3 T/Th 18:00 – 20:00
Location – Baldy 19

Number of credits: 4

Instructor(s) names and contact information:
Dr. Carl Alphonce
e-mail : alphonce@buffalo.edu

On-line appointment: https://calendly.com/alphonce
Office hours: Mondays 10:15 – 11:45
Tuesdays 9:15 – 10:45

COURSE DESCRIPTION

Software is seldom bug-free. Finding and fixing the source of unintended behavior in software can be challenging. This course covers tools and techniques for identifying and locating various types of quality defects in code (such as memory bugs, performance bugs, dependency bugs) and how to write code that lends itself to debugging.

Course Prerequisite: CSE220 Systems Programming

STUDENT LEARNING OUTCOMES

Course Learning Outcome	Program Outcomes / Competencies	Instructional Method(s)	Assessment Method(s)
(I) Employ static and dynamic analysis tools to detect faults in a given piece of software.	CS program: (2) An ability to design, implement, and evaluate a computing-based solution to meet a set of computing requirements in the context of the program's discipline (6) An ability to apply computer science theory and software development fundamentals to produce computing-based solutions CEN program: (6) An ability to develop and conduct appropriate experimentation, analyse and interpret data, and use engineering judgement to draw conclusions (7) An ability to acquire and apply new knowledge as needed, using appropriate learning strategies	Lecture-based instruction Lab-based hands-on exercises, both individual and group	Lab exercises Exploratory projects Lab practical exam Process project
(II) Employ profiling tools to identify performance issues (both time and memory) in a given piece of software.			
(III) Employ testing frameworks to write tests that fail in the presence of software faults, and pass otherwise			
(IV) Employ a structured, methodical approach to detecting, testing, identifying and correcting software faults.			Process project Exploratory projects Lab practical exam
(V) Work productively as a member of a software development team.	CS program: (5) An ability to function effectively as a member of leader of a team engaged in activities appropriate to the program's discipline	Lab-based hands-on group exercises	Process project Exploratory projects

	<u>CEN program:</u> (5) an ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives		

ABET CAC Student Outcome support (CS):

STUDENT OUTCOME	1	2	3	4	5	6
SUPPORT LEVEL	0	1	0	0	2	2

ABET EAC Student Outcome support (CEN):

STUDENT OUTCOME	1	2	3	4	5	6	7
SUPPORT LEVEL	0	0	0	0	2	2	2

<u>COURSE REQUIREMENTS</u>

- TEAM WORK
  - (PRE)/(PST) A "process" team project, done twice, once as a pre-assessment in weeks 1, 2, and 3 of the semester, and a second time as a post-assessment in weeks 10, 11 and 12. Students are required to document their development/debugging process. Learning outcome (IV).
  - (EXP) Team-based exploratory project. This project asks students to apply the tools and techniques they have been taught up to that point in the course to open-source projects. Students must document their use of the tools and the results they obtained. Covers learning outcomes (I), (II), (III) and (IV).
- INDIVIDUAL WORK
  - (LEX) Twice weekly lab-based exercises, completed in the lab session. These are structured to give students practice with the full range of tools and techniques discussed throughout the semester, and so cover learning outcomes (I), (II), and (III). Attendance in lab is mandatory, and counts towards the LEX grade.
  - (LPR) A two-part in-lab practical exam, in week 13. Covers learning outcomes (I), (II), (III) and (IV).
- ENGAGEMENT
  - (ACT) Active learning is incorporated into lecture to promote and support student learning.

<u>GRADING POLICY</u>

- Each piece of student work will be assessed using performance indicators with associated rubrics, with performance levels "insufficient evidence", "developing", "secure", and "exemplary". The overall grade for a piece of work is determined by comparing actual performance relative to performance expectations, published with each assignment. Towards the end of the course students are expected to perform at or above the "secure" level.
- TopHat will be used to administer student response questions. Students earn a point for each question they answer, and additional point for each question they answer correctly.

Component weighting

	Weighting	Assessment / Assignment
Team work	2%	Pre-assessment process project (PRE)
	12%	Exploratory projects (6% each) (EXP)
	16%	Post-assessment process project (PST)
Individual work	30%	Lab exercises (LEX)
	30%	Lab practical exams (LPR)
Mixed	10%	Student Response Questions (ACT)
	100%	TOTAL

Course Grades:

Grade	Quality Points	Percentage
A	4.0	93.0% -100.00%
A-	3.67	90.0% - 92.9%
B+	3.33	87.0% - 89.9%
B	3.00	83.0% - 86.9%
B-	2.67	80.0% - 82.9%
C+	2.33	77.0% - 79.9%
C	2.00	73.0% - 76.9%
C-	1.67	70.0% - 72.9%
D+	1.33	67.0% - 69.9%
D	1.00	60.0% - 66.9%
F	0	59.9 or below

Any work missed for legitimate and documented reasons can be made up, but arrangements must be made with the instructor in a timely fashion (no later than due date, unless medically unable).

Incompletes (I/IU): Unless superseded by changes in university policy, a grade of incomplete ("I") indicates that additional course work is required to fulfill the requirements of a given course. Students may only be given an "I" grade if they have a passing average in coursework that has been completed and have well-defined parameters to complete the course requirements that could result in a grade better than the default grade. An "I" grade may not be assigned to a student who did not attend the course.

ACADEMIC INTEGRITY

Academic integrity is a fundamental university value. Through the honest completion of academic work, students sustain the integrity of the university while facilitating the university's imperative for the transmission of knowledge and culture based upon the generation of new and innovative ideas. See the Academic Integrity Policies of the university (https://catalogs.buffalo.edu/content.php?catoid=1&navoid=19#academic-integrity) as well as the CSE department (https://engineering.buffalo.edu/computer-science-engineering/information-for-students/undergraduate-program/cse-undergraduate-academic-policies/cse-academic-integrity-policy.html for details.

ACCESSIBILITY RESOURCES

Accessibility Resources coordinates reasonable accommodations for equitable access to UB for students with disabilities.

https://www.buffalo.edu/studentlife/who-we-are/departments/accessibility.html

COUNSELING SERVICES

Counseling Services can help with emotional issues, stress, crisis management and much more to support mental wellness through a variety of services.

https://www.buffalo.edu/studentlife/who-we-are/departments/counseling.html

COURSE ORGANIZATION / SCHEDULE (TENTATIVE AND SUBJECT TO CHANGE)

Week #	Topic
1	Overview, "Golden rules" of debugging
2	Compiler (incl. flags), Code repositories (git)
3	Specifications and testing frameworks (unit testing)
4	Debugging process (gdb)
5	Test Driven Development (TDD, behavioral/structural testing), build tools (make)
6	Build tools (make), coverage testing (gcov)
7	Process exercises
8	Performance issues, profiling (gprof)
9	Memory issues (valgrind)
10	Shell scripting
11	Teamwork issues (collaboration boards)
12	Review: memory issues, build tools
13	Review: process
14	Tools in other languages

COURSE MATERIALS

- *The Developer's Guide to Debugging*, Grötker, Holtmann, Keding, Wloka. Springer Science + Business Media B.V., 2008 (recommended)
- *TopHat* student response system (required)

VALUES STATEMENT

The Department of Computer Science and Engineering at the University at Buffalo is a community dedicated to supporting excellence in scholarship and professionalism in all areas of computing. As a community we are bound together by humanity, diversity, equity, inclusiveness, and integrity.

> *Humanity* embodies the ideal that all people are worthy of respect and dignity.
>
> *Diversity* celebrates that every lived experience informs and can give voice to new discoveries, the lifeblood of innovation.
>
> *Equity* recognizes that opportunities must be accessible to all.
>
> *Inclusiveness* ensures that all are welcome and know they are valued members of the CSE community.
>
> *Integrity* is the obligation to earn and maintain the trust of others.

In concert, these ideals are the foundation for effecting positive change in the world and contribute to personal and professional growth and success.