# SFWE 302: Software Architecture and Design
# COURSE SYLLABUS

## Course Description

Software architecture is the blueprint of the system. At the high level, we recognize building blocks and connections between the blocks that follow combination rules. This course teaches students how to use modern processes, methods, and tools used in architecting and designing software systems. Students will learn theory and gain practical experience in developing various architectural styles. Covered topics include motivation for system qualities, architecture definition and documentation, and different types of international and implied dependencies.

## Course Prerequisites

Advanced standing and successful completion of ECE 275 and SFWE 303 (or parallel enrollment)

## Instructor and Contact Information

**Instructor Name**: Tomas Cerny
**Email**: tcerny@arizona.edu
**Office**: Old Engineering Blg 122
**Office Hours**: Teams | TR 1-3pm or upon previous arrangements

*Appointments can be made outside of normal office hours by contacting the instructor(s) to schedule a time that is mutually convenient.*

## Course Objectives

During this course, you will:

1) Discover different architectural styles and impact on quality attributes
2) Explore how various software quality attributes (i.e., reliability, usability, compatibility, interoperability, etc.) can influence the architecture and design of a software product.
3) Leverage appropriate design and technology to realize architectural styles
4) Use Java language to master different architectural style
5) Architect and design a software product for a monolithic or distributed environment

## Expected Learning Outcomes

Upon completion of this course, you should be able to:

1) Develop the architectural, interface, and component design elements for a comprehensive software design that meets specified design requirements

    a. [ABET Student Outcomes 1,2 and 7]
2) Document a software design using Unified Modeling Language (UML) architectural views and design constructs (i.e., sequence diagram, deployment diagram)
    a. [ABET Student Outcome 2,3]
3) Select standard quality attributes to influence a product's software design.
    a. [ABET Student Outcome 4, 6]
4) Develop software architecture for a system proof of concept
    a. [ABET Student Outcomes 1, 2 and 7]
5) Select common architectural styles to design software systems that meet specified quality attributes.
    a. [ABET Student Outcome 6]

## Textbooks & Software

### Required Textbooks (1)

Software Architecture in Practice, 4th Edition is a book written by Len Bass, Paul Clements, and Rick Kazman. It was released in August 2021 by Addison-Wesley Professional. The ISBN is 9780136885979.

The book is 464 pages long and weighs 2.11 pounds. It measures 6.18 x 1.5 x 9.41 inches.

## Modules

**Module 1** – Introduction to Software Architecture
This module introduces this course and its requirements and plans. It also gives definitions of software architecture and related notions. We also introduce Unix.

**Module 2** – Software Qualities
This module introduces software quality and details some of the most important ones.

**Module 3** – Architectural Styles
This module introduces different architectural styles along with their featured qualities.

**Module 4** – Communicating processes.
This module introduces different communication options among independent components.

Assignment - Parsing from CSV files into Collections, Filtering and Sorting Collections

**Module 5** – Documenting Software Architecture
This module introduces architectural views for documenting Software Architecture

**Module 6** – Call and return style - Polymorphisms
This module introduces the call and return style and gives a basic introduction to polymorphism.

**Module 7** – Call and return style – Composition
This module introduces the call and return style and gives a basic introduction to composition.

**Module 8** – Midterm I

**Module 9** – Independent components
This module introduces TCP and client-server architecture.

**Module 10** – Data-Flow architecture
This module introduces data flow architecture and concurrency problems.

**Module 11** – Data-centric architecture
This module introduces data-centric style and databases.

**Module 12** – Call and return style – Component-based development
This module introduces component-based development through Spring framework.

**Module 13** – Implicit Invocation
This module introduces implicit Invocation and messaging systems.

**Module 14** - Exam

## Required Software

You will either need to install or create an account for the following software – links and instructions are available in the ***Start Here*** module of the D2L course site.

- **Integrated Development Environment (IDE) -** You will need a Java IDE of your choice for code development.
- **Unix/Linux** – Install Dual boot or Virtual Machine
- **Docker** – install docker
- **VisualParadigm -** Install community edition.

## Assignments and Examinations

### Homework

There will be regular homework assignments on the topics covered in class, with approximately thirteen homework assignments to be completed individually (not on a team basis). There will also be module-based discussion board prompts that each student is required to participate in and will be graded for.

### Knowledge Checks

There might be online self-assessed quizzes, and knowledge check questions with the completion of each module. There will be approximately thirteen knowledge checks that will be factored into your grade. Knowledge checks are an individual assessment of your understanding of the concepts and knowledge covered in a given module.

### Midterms/Exam

There will be two midterms/exams. All will be comprehensive and timed, administered by proctored software online (for online students), and will be available during the class or regularly scheduled exam time. Online students will be provided with information on how to access the exam on the specified date in the course calendar. There will be no alternative exam and scores will be averaged for properly excused absences.

## Grade Distribution, Scale & Policies

The grading distribution for course assignments is as follows:

| | |
|---|---|
| Assignments/Labs: | 70% |
| Midterm II: | 15% |
| Exam: | 15% |
| **Total** | **100%** |

### Grading Scale

The following scale will be used to award final grades for undergraduate students.

| | |
|---|---|
| A | 90-100% |
| B | 80-89% |
| C | 70-79% |
| D | 60-69% |
| E | less than 60% |

### Incomplete (I) or Withdrawal (W):

Requests for incomplete (I) or withdrawal (W) must be made in accordance with University policies, which are available at http://catalog.arizona.edu/policy/grades-and-grading-system#incomplete and http://catalog.arizona.edu/policy/grades-and-grading-system#Withdrawal respectively.

## Student Appeals Policy

The instructor will make every attempt to provide timely feedback on all assignments, knowledge checks, exams, and projects.

You can dispute any grade that you receive within three days days from the time the grade was awarded. If you feel that you have received an unfair assessment of your performance on any given homework assignment, knowledge check, exam, or final project, please reach out to the course instructor either in person, via email, or over Zoom as soon as possible. Be prepared to provide substantiated claims for your dispute, including any evidence that would support a re-evaluation of your grade.

## Late Work Policy

Homework/ Knowledge Checks/ Projects: All homework, knowledge checks, and elements of the semester project are due at the time that is specified in the course schedule and/or D2L content pages. *Late homework and projects will not be accepted without prior approval by the instructor and will receive 0 points.*

Exams: If a student misses a midterm/exam, the score will be calculated as an average from the other two midterms/exam. Students cannot miss more than one midterm to earn a passing grade. Please note that students must contact the instructor well in advance and provide *written* documentation for the reason that he/she will not be able to attend the regularly scheduled exam. It is up to the discretion of the instructor to accept the justification provided by the student.

## Code of Academic Integrity and Honor Code

Any violation of academic integrity will result in an F in the course; students are advised to self-assess them at https://deanofstudents.arizona.edu/policies/code-academic-integrity
Plagiarism or any form of cheating involves a breach of student-teacher trust. This means that any work submitted under your name is expected to be your own, neither composed by anyone else as a whole or in part nor handed over to another person for complete or partial revision. Be sure to document all ideas that are not your own. Instances of plagiarism or any other act of academic dishonesty will be reported to the Honor Council and may result in failure of the course. Not understanding plagiarism is not an excuse. I expect you, as a Baylor student, to be intimately familiar with the Honor Code at
https://law.arizona.edu/sites/default/files/Honor_Code_2015.pdf
Few examples (read well):

- Using a source fragment other than the course textbook, the course website, or your professor to obtain credit for an assignment, project, or exam.
- Copying another student's work. Simply looking over someone else's source code is copying.
- Providing your work for another student to copy.
- Collaboration on any assignment, unless the work is explicitly given as collaborative work. Any discussion of an assignment or project is considered collaboration.
- Using notes or books during an exam.
- Giving another student answers during an exam.
- Reviewing a stolen copy of an exam.
- Plagiarism.
- Studying tests or using assignments from previous semesters, except when provided by the professor.
- Providing someone with tests or assignments from previous semesters.
- Taking an exam for someone else.
- Turning in someone else's work as your own work.
- Studying a copy of an exam prior to taking a make-up exam.
- Providing a copy of an exam to someone who is going to take a make-up exam.
- Giving test questions to students in another class.
- Reviewing previous copies of the instructor's tests without permission from the instructor.

## Course Behavior Policy

To foster a positive learning environment, students and instructors have a shared responsibility. We want a safe, welcoming, and inclusive environment where all of us feel comfortable with each other and where we can challenge ourselves to succeed. To that end, our focus is on the tasks at hand and not on extraneous activities (e.g., texting, chatting, reading a newspaper, making phone calls, web surfing, etc.).

## University Policies

Links to the following UA policies are available at, https://academicaffairs.arizona.edu/syllabus-policies:

- Absence and Class Participation Policies
- Threatening Behavior Policy
- Accessibility and Accommodations Policy
- Code of Academic Integrity
- Nondiscrimination and Anti-Harassment Policy
- Subject to Change Statement