

CSE 4361/5322 SOFTWARE DESIGN PATTERNS

August 13, 2025

1 General

Lectures:	MWF 9:00AM-9:50AM, NH 110 or on TEAMS
Instructor:	David C. Kung, ERB 532
Office Phone:	817-272-3785
Email:	k u n g AT u t a DOT e d u
Faculty Profile:	https://www.uta.edu/academics/faculty/profile?user=kung
Office Hour:	
In-person days:	10:00 AM-11:00 AM MWF (ERB 532)
Online days:	10:00 AM-11:00 AM MWF (TEAMS)
Or: By appointment	
GTA:	Imtiaz Bin Rahim; ixb6394@mavs.uta.edu
GTA Office:	ERB 512
GTA Office Hour:	TBA

2 Course Description

CSE 4361. SOFTWARE DESIGN PATTERNS (3-0). In-depth study of software design patterns including description of patterns, design principles and techniques used by patterns as well as application of patterns to solving practical design problems. Team project. Prerequisites: Admitted into an Engineering Professional Program. C or better in CSE 3311.

3 Time and Place of Class Meetings

This course operates on Central Time. All times listed for class meeting times, exams, and assignment deadlines are in Central Time (CT).

Classes meet in NH 110 on the in-person days, and meet on TEAMS on the online days. The in-person days and online days are shown in the course schedule table in Section 1, where the green slots are online and the white slots are in-person.

4 Classroom/Lecture Recording Policy

Faculty maintain the academic right to determine whether students are permitted to record classroom and online lectures. Recordings of classroom lectures, if permitted by the instructor or pursuant to an ADA accommodation, may only be used for academic purposes related to the specific course. They may not be used for commercial purposes or shared with non-course participants

except in connection with a legal proceeding.

Recording of classroom and online lectures in this course is allowed.

Both in-person and online lectures are recorded. In particular, in-person lectures are recorded on Echo 360 and available on Canvas. Online lectures are recorded on TEAMS.

5 Student Learning Outcomes

By the end of this course, you will be able to:

1. know the controller, expert, creator and the 23 Gang of Four design patterns;
2. know the design problems solved by each design pattern;
3. know the benefits and liabilities of each design pattern;
4. apply patterns to solve design problems during the software development life cycle.

6 Reference Books

1. David Kung, “Software Engineering,” 2nd Edition, McGraw-Hill 2023. ISBN: 978-1260792683. This is the 2nd edition of the following book.
2. David Kung, “Object-Oriented Software Engineering: An Agile Unified Methodology,” McGraw-Hill 2013. This is the old edition.
3. Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, “Design Patterns: Elements of Reusable Object-Oriented Software,” Addison-Wesley, 1995.

7 Tentative Schedule

See Figure 1.

Date	Reading	Class Activity/Topics	Assignment (due date)
8/18-8/22	2, 5, 7-11	Syllabus, background survey (on Canvas). Team project, teams formed by instructor & GTA. Review of agile unified methodology (AUM).	Background survey (8/18)
8/25-8/29	2, 5, 7-11	Review of agile unified methodology (AUM). Controller, expert and creator patterns.	
9/1-9/3-5	16	Labor Day Holiday (no class). Intro to software design patterns, <i>singleton, adapter</i> . Techniques used by patterns.	
9/8-12	16	A process for applying patterns. <i>Composite, iterator, flyweight (3 UC1 patterns)</i> .	Team: UC1 Edit Booth Floor Plan (due 9/28 11:59 PM)
9/15-19	16	<i>Bridge, command, template method, factory method</i> .	
9/22-26	16	Visitor (1 UC2 pattern), proxy, prototype.	Team: UC2 Generate Statistics Report (due 10/14 11:59 PM).
9/29-10/3	17	UC1 class exercise (domain modeling, actor-system interaction modeling, object interaction modeling, applying patterns, design class diagram, coding).	Individual: UC1 Team Member Evaluation (due 10/10 11:59 PM).
10/6-10	13, 16	<i>Interpreter, chain of responsibility (3 UC3 patterns including the command pattern)</i> .	Team: UC3 Edit Discount Policy (due 10/26 11:59 PM).
10/13	2, 5, 7-11, parts of 16-17	Midterm exam (AUM, intro to patterns, tech used by patterns, a process for applying patterns, patterns learned so far).	
10/15-17		UC2 class exercise.	Individual: UC2 Team Member Evaluation (due 10/24 11:59 PM).
10/20-10/24	13, 16, 22	<i>Observer, state (2 HW patterns)</i>	Individual homework (HW): Automatic lawn mower (due 11/9 11:59 PM).
10/27-31		UC3 class exercise.	Individual: UC3 Team Member Evaluation (due 11/7 11:59 PM).
11/3-11/7		<i>Mediator, memento, strategy</i> .	
11/10-11/14		HW1 class exercise (same as UC1 class exercise).	
11/17-11/21	16	<i>Façade, object adapter, decorator. Abstract factory, builder</i> .	
11/24		UTA no class schedule (no class for this course).	
11/26-11/28		Thanksgiving Holiday (no class). Thanksgiving Holiday (no class).	
12/1		Review for final exam.	
12/2		Last day of classes (no class).	
12/5		Final exam 2:00 PM – 4:30 PM in classroom (TBD)	

Note: Green slots are online, white slots are in-person sessions.

Figure 1: Tentative schedule

Note: the chapters shown in the Readings column of Figure 1 refer to reference book 1 or 2 (their chapters have the same chapter numbers).

8 Workload and Distribution

Work Item	Description	Unit %	Subtotal
Team project	3 use cases	15%	45%
Individual Homework	1 assignment	15%	15%
Midterm exam	1 midterm exam	10%	10%
Final exam	1 final exam	20%	20%
Class attendance		10%	10%
Course total			100%

- One semester team project (45%):

The project requires the teams to perform analysis, design and implementation of three use cases of a real-world application. Teams are required to apply specific patterns to the design and implementation of the use cases. For each of the use cases, the team is required to do and submit the following:

1. Produce a description of the business process for the use case.
2. Do domain model brainstorming on the description produced above, classify the brainstorming results into classes, attributes of classes and relationships between classes, and visualize the classification results in a UML class diagram (which must not show any operations).
3. Specify the high-level use case (that is, when and where the use case begins and when the use case ends).
4. Specify the expanded use case (that is, how an actor will interact with the system to carry out the foreground processing of the use case). Identify the nontrivial step(s) of the expanded use case.
5. Apply each of the required Gang of Four (GoF) patterns by copying the structural design of the pattern, and replacing the generic class names, method names and attribute names with application-specific classes, methods and attributes.
6. For each of the nontrivial steps of the expanded use case produce a scenario description to describe how software objects interact with each other to process the actor request to produce the system response (as specified in the nontrivial step). The scenario description must also apply the required design patterns including the controller, expert and creator patterns whenever applicable.
7. Convert the scenario description to an informal sequence diagram in which the messages between the objects are labeled by English texts.
8. Convert the informal sequence diagram to a design sequence diagram, which is the same as the informal sequence diagram except that the messages between the objects are converted to function calls with parameters and parameter types, and possibly return values and return types.
9. Derive a design class diagram (DCD) from the design sequence diagram. That is, derive classes, methods and attributes of classes, and relationships between the classes.
10. This is optional but if you do this, your team will get 15% bonus. Implement all classes and all methods in the DCD in a programming language of your choice. Compile, run

and debug the software. Produce screen shots showing the working of the software. Submit the code, executable, and screen shots.

To ensure that team members attend team meetings and actively participate in team discussion, all teams are required to use Microsoft TEAMS to record team meetings so that the instructor and the GTA can check each team member's participation in the project work.

- One individual homework assignment (15%).
- One open-book midterm exam (10%). It is not open-note, and no electronic devices are permitted.
- One open-book final exam (20%). It is not open-note, and no electronic devices are permitted.
- Class attendance (10%). Students are required to attend all classes in-person or online as scheduled. If a student cannot attend a class, then the student must inform the instructor and the GTA in advance and before the class time. If the student cannot inform the instructor and the GTA in advance due to medical reason, the student must provide a doctor's letter. Fail to do so will result in proportional deduction of point in this category.

9 Grade Distribution

Total Score	≥ 85	≥ 70	≥ 60	≥ 50	< 50
Grade	A	B	C	D	F

The grades are computed by a program according to your scores. If you get 84.95 then you will get a "B", not an "A" even if the score is so close to 85.

10 Team Member Evaluation Form

Each student is required to submit a team member evaluation form (see next page) on Canvas after the team submission of each use case (there are three submissions of use cases for this semester). Use this form to appraise those team members that you feel their contributions should be credited and provide the instructor information about team members who need improvement. We will keep this confidential. Students who fail to submit this form will lose 1% for the use case team work.

Students' performance in the team is taken into consideration. Each negative point in the peer evaluation deducts one point from your iteration score. So if a student gets -2 for each of the six categories from one peer, then his score for that iteration will be reduced by 12 points.

Project Team Member Evaluation Form

Team#___ Iteration#____ Course#____ Fall / Spring Year____

Please submit hardcopy or fax to David Kung 817-272-3784, EMAIL NOT ACCEPTABLE

Most team members perform well in a project team. However some members perform extremely well and some very poorly. It is constructive to encourage the outstanding members and inform those who need improvements. This form allows you to convey such information to your team members whenever you deem there is such a need.

Please give an integer rating of -2 (poor), -1 (below average), 0 (average), +1 (above average), or +2 (excellent) for some of the aspects of the members you want to convey your assessment. Your evaluation might be reproduced (to hide your identity) and presented to the relevant members. However, the identity of the evaluator will be kept absolutely confidential in all cases.

Member name					
Group meeting attendance					
Group discussion					
Individual assignment					
Technical contribution					
Organizational contribution					
Overall performance					

Comments: (use additional sheets if needed)

Name: _____ Signature: _____ Date: _____

Figure 2: Team member evaluation form

11 Project and Assignment Grading Criteria

- 1) Correctness – the solution adequately solves the given problem
- 2) Soundness – the solution is well justified
- 3) Efficiency – the solution is among the simplest ones possible
- 4) Organization – the presentation of the solution is easy to understand and logically organized
- 5) Clarity – the solution is clearly stated and tables and figures are professionally produced
- 6) Grammar, spelling, and writing – correct grammar and spelling, and legible writing

1) – 2) are worth about 60% of the weight and 3) - 6) about 40%.

12 Assignment Rules

1. Late submissions up to one week are subjected to 10% deduction. Submissions passed one week will not be accepted and counted as no submission.

2. You can discuss homework with your classmates but you are not allowed to copy the solutions from or share the solutions with anybody. If you violate this rule, then you will receive no credit for that assignment unless you can prove that you are not involved.
3. The GTA will do most of the grading. If you do not agree with the result, contact the GTA first. Please contact the instructor if you cannot reach a consensus. This would help the GTA improve her/his grading skill and avoid inconsistency due to improper interference of the instructor.
4. No additional make-up assignment will be provided for any student to improve grade.

13 Class Email Alias

Messages to students will be sent to a mailing list. If you do not receive such email, please contact the instructor as soon as possible. You should also contact me when your university email account has changed.

14 Your Standing and Class Statistics

After each assignment or test has been graded, the TA will distribute to each of you your scores and grade up to that assignment or test. You will also receive class performance statistics. The TA is required to timely distribute these to you. Please feel free to inquire the TA or me if you do not receive these in due time.

15 Request for Early Leave

Requests for permission to go home before the final exam date will never be granted except for medical reasons and with a proof from a doctor. Students who do not participate in the final exam will not receive the scores for the final exam except that the final exam is waived

16 SE Code of Ethics and Professional Practice

ACM/IEEE Software Engineering Code of Ethics and Professional Practice

For the full version, see <http://www.acm.org/serving/se/code.htm#full>.

Software engineers shall commit themselves to making the analysis, specification, design, development, testing and maintenance of software a beneficial and respected profession. In accordance with their commitment to the health, safety and welfare of the public, software engineers shall adhere to the following Eight Principles:

1. PUBLIC - Software engineers shall act consistently with the public interest.

2. CLIENT AND EMPLOYER - Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest.
3. PRODUCT - Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.
4. JUDGMENT - Software engineers shall maintain integrity and independence in their professional judgment.
5. MANAGEMENT - Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6. PROFESSION - Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7. COLLEAGUES - Software engineers shall be fair to and supportive of their colleagues.
8. SELF - Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession.