EN.605. 707 section 81 Syllabus

Software Patterns

Course Information

Software Patterns

EN.605. 707 81 (3.0 Credits)
Fall 2021 [AE Fall 2021]

Description

Software patterns encapsulate the knowledge of experienced software professionals in a manner that allows developers to apply that knowledge to similar problems. Patterns for software are analogous to the books of solutions that enable electrical engineers and civil engineers to avoid having to derive every new circuit or bridge design from first principles. This course will introduce the concept of software patterns, and explore the wide variety of patterns that may be applied to the production, analysis, design, implementation, and maintenance of software. The format of the course will emphasize the discussion of patterns and their application. Each student will be expected to lead a discussion and to actively participate in others. Students will also be expected to introduce new patterns or pattern languages through research or developed from their own experience. Programming exercises performed outside of class will be used enhance discussion and illustrate the application of patterns.

Department: PE Computer Science
College: Engineering and Applied Science Programs for Professionals

Instructor

Jon Lindberg
✉ jlindbe2@jhu.edu

Communication Policy :

Work: 240-228-4737

I prefer that students contact me via email. Please be sure to include course number in the subject line. I will make every effort to respond to your inquiry within 1 hour on weeknights and 2 hours on weekends and during business hours.

Office Hours :

This course will use Zoom to facilitate weekly, synchronous office hours. You are not required to participate in office hours; however, you may find them very beneficial for receiving more timely answers to questions related to the course content and assignments. Special attention will be given during office hours towards topics related to implementing the assignments.

During the first week of the course I will conduct a student survey to determine the best day and time of the week to schedule the office hours. Once the day and time have been determined, I will set up office hours links within the Calendar. Students will click that link to access Adobe Connect and participate in the office hours. Recorded office hour sessions will be posted to the announcements area for any students who were unable to participate in the "live" sessions or for students who like to listen to them again. Office hours will end 15 minutes after the last student has disconnected from the session.

Course Structure :

The course materials are divided into modules which can be accessed by clicking Course Modules on the left menu. A module will have several sections including the overview, content, readings, discussions, and assignments. You are encouraged to preview all sections of the module before starting. Modules run for a period of seven (7) days. You should regularly check the Calendar and Announcements for assignment due dates. A final project is assigned at the beginning of the semester and is due at the beginning of the final module of the course.

Course Topics :

Factory Method/Strategy/ Decorator
Composite/Iterator/ Refactoring

Template Method/ Abstract Factory/Builder
Singleton and Monostate/Proxy
Idioms and Pattern Languages
Adapter/Bridge
Mediator/Observer
Chain of Responsibility/ Memento
Analysis Patterns, Architectural Patterns and Antipatterns
Prototype/Command
State/Visitor
Flyweight/Interpreter/ Façade
Championing Patterns and Final Project

Course Goals :

To recognize the use and applicability of software patterns and employ that knowledge to create solutions that increase the flexibility and maintainability of software. Understand the different types of patterns that apply to the stages of the software engineering process. Experience the tradeoffs between pattern-based and non-pattern-based solutions and the consequences of each pattern.

Course Learning Outcomes (CLOs) :

Understand the form and content of design patterns, what is not included in the patterns and how they fit into object oriented software design.

S5862

Gain familiarity with the other types of patterns available including analysis patterns, architectural patterns and anti-patterns.

S5863

Determine when it is appropriate to use patterns.

S5864

Develop a deeper understanding of the fundamental concepts of object oriented programming and how patterns illustrate and reinforce them.

S5865

Develop the ability to think criticality about the structure of software.

S5866

Required Text and Other Materials

Textbooks :

Gamma, E., et. al. (1995). Design Patterns: Elements of Reusable Object-Oriented Software. New York, NY: Addison-Wesley

ISBN 0-201-63361-2.

Textbook information for this course is available online through the appropriate bookstore website: For online courses, search the MBS website at http://ep.jhu.edu/bookstore.

Other Materials & Online Resources :

Other course readings are available through the eReserves link on Blackboard.

Required Software :

Students will need access to a Java, C++, or Swift compiler and whatever software development environment they are comfortable using. No particular version of either language is required, though students using C++ should also download a copy of the Boost libraries (www.boost.org) to make use of the regex library. Students should also have the ability to create and read zip files for packaging assignments to turn in. Students may use another language than Java, C++, or Swift to implement the assignments with permission of the instructor, though students will be required to adapt the assignment framework to that language on their own.

Technical Requirements :

You should refer to **Support** on the course menu for a general listing of all the course technical requirements.

Evaluation and Grading

Student Coursework Requirements :

It is expected that each module will take approximately 7–10 hours per week to complete. Here is an approximate breakdown: reading the assigned sections of the as well as some outside reading text (approximately 2-3 hours per week), listening to the audio annotated slide presentations (approximately 1 hour per week), and implementing programming assignments (approximately 3-4 hours per week), participating in course discussions (approximately 1-2 hours per week).

This course will consist of four basic student requirements:

1. **Preparation and Participation (Module Discussions)** (20% of Final Grade Calculation)

You are responsible for carefully reading all assigned material and being prepared for discussion. The majority of readings are from the course text. Additional reading will be assigned to supplement text readings. The initial discussion questions are provided for read-ahead in each module, but you are encouraged to pose your own questions related to the pattern and its use in the discussion threads. Remember also to participate in the discussion questions posted by the Discussion Leaders for the module.

A new question for each pattern will be posted each of five consecutive days during the module. Post your response to the question or to other students' responses and ask follow-up questions. While everyone will not be able to respond to every question, it is expected that each student will participate in the discussion of most questions during each module. Discussion on early questions may continue throughout the module.

Part two of your grade for module discussion is your interaction (i.e., Critical Thinking). Just posting your response to a discussion question is not sufficient; we want you to interact with your classmates. Be detailed in your postings and in your responses to your classmates' postings. Feel free to agree or disagree with your classmates or the instructor. Include insight from the text, the related pattern readings, and your experience implementing the pattern either in class or elsewhere. Please ensure that your postings are civil and constructive.

I will monitor and participate in the module discussions, but discussion should be directed primarily to the class rather than directly to me. It is the responsibility of the student with Discussion Leadership requirement for a pattern to summarize the overall discussions and post the summary for the module in the course Wiki. I will fill this role for patterns which are not chosen and for the patterns in Modules 1 and 2.

Evaluation of preparation and participation is based on contribution to discussions.

Preparation and participation is evaluated by the following grading elements:

1. Timeliness (20%)
2. Breadth of Participation (30%)
3. Critical Thinking (50%)

Preparation and participation is graded with the formula (2T+3B+5C)/3 and the following point values:

Discussion Grading – Points accumulate to 10 for each assignment.				
Points	0	1	2	3
Critical Thinking	Statements contribute no new ideas, have minimal depth & detail. Rehashes or summarizes others. Unclear if student grasps topic. Mostly agrees or asks questions.	Contributes few new ideas. Statements contain some rehashing or summary of others. Seems to grasp discussion topic at least superficially.	Statements offer some depth and insight and some new ideas. Shows an understanding of the discussion topic.	Statements are consistently in depth and detailed with new, insightful ideas clearly connected to the topic. Demonstrates a command of the discussion topic.
Breadth of Participation	Contributes to one discussion.	Contributes one or two discussions and the discussion leadership.	Contributes to a majority of discussions including discussion leadership.	Contributes to all discussions.
Timeliness	No posts until near due date.	First post in second half of module. No further participation.	First post within first three days of module. No further participation.	First post within first three days of module. Continued participation for remainder.

2. **Assignments** (30% of Final Grade Calculation)

Assignments generally require the development of a software solution implementing each of the patterns being discussed in the module. A small number of assignments may be completed by clearly describing how the existing implementation in the provided software framework already fully realizes the pattern, identifying which objects, methods and relationships fulfill all of the roles in the pattern to receive all 10 points for the pattern. One or two of the assignments may be completed by providing a strong and convincing argument for why the requested implementation of the pattern is not appropriate with reference to the pattern's consequences.

All assignments are due according to the dates in the Calendar.

Late submissions will be reduced by one letter grade for each week late (no exceptions without prior coordination with the instructors).

If, after submitting an assignment you are not satisfied with the grade received, you are encouraged to redo the assignment and resubmit it. If the resubmission results in a better grade, that grade will be substituted for the previous grade, but treated as if it were submitted late.

Each pattern in the assignment will be rated by the following points, and all patterns in the assignment averaged together.

Assignment Grading – Points accumulate to 10 for each assignment.				
Points	0	1	2	2.5
Structure	Structure does not match the canonical form of the pattern or of any reasonable alternative.	Structure deviates significantly from the canonical form in a way that does not preserve the intent of the pattern.	Structure follows the canonical form to a large part, but is missing one or more important features without justification.	Structure faithfully follows the canonical form of the pattern or of an alternate form that strictly matches the intent of the canonical solution.
Clarity	Roles of objects, classes, methods and relationships participating in the pattern are not clearly identified.	Some roles are identified, but it is not clear how they relate to others.	Most roles are evident.	All roles in the pattern are clearly identified.

Assignment Grading – Points accumulate to 10 for each assignment.				
Appropriateness	The pattern is used in place where it does not fit, or where there is an existing implementati on already.		The pattern is used to solve an appropriate problem, but ignores a pre-existing implementation.	The pattern's use is completely appropriate to the circumstances.
Functionality	The implementati on could not be made to work as attempted, even with an indefinite time for debugging.	The implementation is correct in the broadest sense, but has significant flaws. For example, there is an abstract Strategy and concrete Strategies, but there is no Context, or the Context has inappropriate knowledge of the concrete Strategies.	The implementation is mostly correct, but omits one or more important details, especially with regard to issues raised in the discussion but not mentioned by the text.	The implementation fully implements the pattern and any attendant details, e.g., initialization of Factory Method.

3. **Final Project** (30% of Final Grade Calculation)

The final project is a paper on one of two topics, either presenting an analysis of the use of patterns or potential for pattern use in a piece of open source software, or the documentation of an original pattern. For the first option the software must meet the following requirements:

- It must be open source. I should be able to find the source online myself.
- It must not be written by any of the following people: yourself, friends, co-workers, fellow students, the instructor.
- The software must be written in a language that the instructor can understand: C, C++, C#, Java, Ada, csh, Executable UML. I'll consider others like LISP or Scheme but prefer not.
- The software must be sufficiently complex to support a rich set of patterns. What usually works best is to find a medium size program and to focus on one part of it.

The project is evaluated by the following grading elements:

1. The architecture of the software (UML diagrams are encouraged). This is a description of the overall purpose of the software, the high-level structure, and where in that structure is the software containing the patterns that you will discuss. (10% of the content)
2. Patterns encountered in the software or that should be applied to the software. This should include patterns at all levels from architecture to idiom. It is preferable to concentrate on 3-5 patterns and treat them in depth than to gloss over many. (50%) Discussion should include:
    1. Implementation topics discussed in class such as the boundary conditions of the pattern (e.g., how is a Strategy initialized or when is a Singleton destroyed)
    2. Differences from the canonical implementation
    3. If the structure is similar to another pattern, why it is one vs. the other
    4. How coupling is reduced and cohesion increased by the pattern's implementation
    5. How modularity and flexibility are improved by the pattern
3. Appropriateness of the patterns, i.e., was the pattern used appropriately in the software, matching the intent given by the GoF. Or if you are discussing the application of patterns to the software, justify their use. (40%)

The pattern analysis form of the final project is graded using the formula (Arch+5Patterns+4App)/3 with points assigned as follows:

Project Grading – Points accumulate to 10 for each assignment.				
Points	0	1	2	3
Architecture	Superficially describes the software focusing mostly on how the software is used instead of on its structure	Describes the software but focuses on details unimportant to understanding the use of the patterns identified	Describes the software purpose and overall structure but does not identify where the patterns fit	Completely and concisely describes the software purpose and structure in relation to the selected patterns

Project Grading – Points accumulate to 10 for each assignment.				
Patterns	No more than one pattern other than Singleton or Strategy is identified. Few aspects of the pattern implementation are addressed. Pattern roles are not identified.	Less than three different patterns are identified after discounting Singleton and Strategy. Some aspects of the pattern implementation discussed in class are addressed. Some of the patterns' roles are correctly identified.	Fewer than five different patterns are identified, or five patterns include both Singleton and Strategy. The critical aspects of the pattern implementation are addressed. Most of the patterns' roles are correctly identified.	Five different patterns are identified only one of which is Singleton or Strategy. Most aspects of the pattern implementation discussed in class are addressed. All of the patterns' roles are correctly identified.
Appropriateness	No critical judgement of the software author's use of patterns is demonstrated.	One or two patterns are a strong fit.	Most patterns identified are a strong fit for the problem they solve, but inappropriate pattern solutions are not addressed.	Each pattern identified is a strong fit for the problem it is solving with substantiating reasoning, or the description clearly identifies why it is not.

For the second option any type of pattern may be described of any type, design, analysis, architecture, anti-, etc. The following criteria must be met:

- **The pattern must be original. Due diligence must be performed to make sure that the name is not used by any existing pattern, even if the patterns are not the same.**
- **The pattern description should include each of the following components: name, intent, motivation, applicability, solution with structure and participants/roles and collaborations, positive and negative consequences, implementation notes, sample code, known uses, and related patterns.**
- **The known uses should be working systems or systems in development. They need not be publicly available, but should not be solutions created solely to justify the pattern. There should be at least two known uses, preferably three.**
- **There is no preferred length for the pattern; reference the variety of lengths of the patterns in the text.**

The pattern creation form of the final project is graded using the formula (A+B+C+3D+3E+2F+2G+2H+I)/4.8 with points assigned as follows:

Project Grading – Points accumulate to 10 for each assignment.				
Points	0	1	2	3
A. Intent	Intent is missing	Requires more than two sentences to describe the pattern, or is ambiguous		Clearly and succinctly describes purpose of pattern
B. Motivation	Motivation is missing	Describes an inapplicable situation	Describes a contrived situation	Describes a common situation that the pattern fits
C. Applicability	Applicability only restates the intent or motivation	Describes the problem being solved but not the specific qualities, for example describing Strategy as being applicable when there are multiple algorithms	Partially describes the specific qualities of the problem solved by the pattern	Describes the specific nature of the problem to which the pattern applies, for example describing Strategy as being applicable when there are multiple algorithms distinguished by the type of object that can be changed without affecting the client
D. Solution	Omits one or more of the roles, structure, or collaborations	Includes all three aspects of the solution but omits multiple necessary details	Includes all three aspects but omits a few details	Clearly identifies all roles, relationships, and collaborations

Project Grading – Points accumulate to 10 for each assignment.				
E. Consequences	Has only positive or negative consequences	Both consequences are described only one or fewer are non-trivial	Both positive and negative consequences are described but not all of them are relevant or non-trivial	Gives multiple positive and negative consequences
F. Implementation	No implementation considerations are described	Few or trivial considerations are described		Describes each of the design tradeoffs that must be considered when implementing the pattern
G. Sample Code	No sample code provided	Sample code is incomplete or does not match motivation	Sample code “works” but does not employ all of the elements of the solution described by the pattern	Simple software implementation applied to the situation described in the motivation
H. Known Uses	No known uses are identified	Known use is retrofitted to justify the pattern	Only one known use is identified	Identifies three specific places where the solution described in the pattern occurs
I. Related Patterns	No related patterns are iidentified	One or more patterns are identified but with a tenuous relationship to the new pattern	Only one relevant related pattern identified	Identifies at least two patterns that aid in implementing the new pattern or which are themselves aided by the new pattern

4. **Discussion Leadership** (20% of Final Grade Calculation)

Each student must provide supplementary materials and questions to support leadership of the discussion of one pattern during the semester. If there are more students than patterns available to lead, some duplication will be permitted.

There are three parts to leading a discussion.

- Real world example, using props or video. This example should be different from the one provided in the paper from AG Communications.
- Software example of the pattern drawn from an actual program design or implementation.
- Providing at least one additional question to motivate understanding of the pattern.

Discussion leadership is graded as follows:

Discussion Lead Grading – Points accumulate to 10 for each assignment.				
Points	0	1	2	3.3
Real World Example	The real world example is a poor analogy for the pattern. It is not clear what roles the pattern components play in the example.	The real world example is a close analogy to the pattern, but differs in one or more important aspects that are not called out by the discussion.	The real world example is a very close analogy to the pattern, but one or more of the abstract parts of the pattern are missing.	The real world example is a very close analogy or counter example for the pattern, and all roles described for the pattern are captured.

Discussion Lead Grading – Points accumulate to 10 for each assignment.				
Software Example	The software example does not include the pattern or uses some other pattern instead.	The software example is appropriate, but is significantly incomplete.	The software example is appropriate, but some roles of the pattern are missing and not addressed by the discussion.	The software example is a complete expression of the pattern.
Demonstration of Understanding	Student is unable to answer nontrivial questions about the pattern and poses no questions not found in the discussion notebook.	Student is able to answer trivial questions about the pattern, but not questions requiring deeper insight. No new questions are posed.	Student is able to answer most questions about the pattern, and asks the class at least one insightful question.	Student is able to answer nearly all questions and requires little or no assistance from the instructor. One or more insightful new questions is presented to the class.

Grading Policy :

Grading

Assignments are due according to the dates posted in your Blackboard course site. You may check these due dates in the Course Calendar or the Assignments in the corresponding modules. I will post grades one week after assignment due dates.

I generally do not directly grade spelling and grammar. However, egregious violations of the rules of the English language will be noted without comment. Use of grammar or spelling that makes understanding of the work difficult or impossible will affect the grade.

A grade of A indicates achievement of consistent excellence and distinction throughout the course—that is, conspicuous excellence in all aspects of assignments and discussion in every week.

A grade of B indicates work that meets all course requirements on a level appropriate for graduate academic work. These criteria apply to both undergraduates and graduate students taking the course.

100–98 = A+
97–94 = A
93–90 = A–
89–87 = B+
86–83 = B
82–80 = B–
79–77 = C+
76-73 = C
72-70 = C-
69-67 = D+
66-63 = D
<63 = F

Final grades will be determined by the following weighting:

Item	**% of Grade**
Weekly programming assignments	30%
Discussion Leadership	20%
Final Project	30%
Participation in class discussion	20%

Policies

Additional Resources :

Personal Wellbeing

If you are struggling with anxiety, stress, depression or other mental health related concerns, please consider connecting with the Johns Hopkins Student Assistance Program (JHSAP). If you are concerned about a friend, please encourage that person to seek out our services. JHSAP can be reached at 443-287-7000 or https://jhsap.org/

Tutoring Website

Johns Hopkins Engineering for Professionals offers a tutoring connection network that allows students to connect with other Johns Hopkins Engineering students or alumni for tutoring services. This service allows students to search a list of courses to “Find a Tutor” or complete a profile to “Become a Tutor.” More information about this service can be found on the tutoring website (https://tutor.ep.jhu.edu/).

Deadlines for Adding, Dropping and Withdrawing from Courses

Students may add a course up to one week after the start of the term for that particular course. Students may drop courses according to the drop deadlines outlined in the EP academic calendar (https://ep.jhu.edu/student-services/academic-calendar/). Between the 6th week of the class and prior to the final withdrawal deadline, a student may withdraw from a course with a W on their academic record. A record of the course will remain on the academic record with a W appearing in the grade column to indicate that the student registered and withdrew from the course.

Academic Misconduct Policy

All students are required to read, know, and comply with the Johns Hopkins University Krieger School of Arts and Sciences (KSAS) / Whiting School of Engineering (WSE) Procedures for Handling Allegations of Misconduct by Full-Time and Part-Time Graduate Students.

This policy prohibits academic misconduct, including but not limited to the following: cheating or facilitating cheating; plagiarism; reuse of assignments; unauthorized collaboration; alteration of graded assignments; and unfair competition. Course materials (old assignments, texts, or examinations, etc.) should not be shared unless authorized by the course instructor. Any questions related to this policy should be directed to EP’s academic integrity officer at ep-academic-integrity@jhu.edu.

Students with Disabilities - Accommodations and Accessibility

Johns Hopkins University values diversity and inclusion. We are committed to providing welcoming, equitable, and accessible educational experiences for all students. Students with disabilities (including those with psychological conditions, medical conditions and temporary disabilities) can request accommodations for this course by providing an Accommodation Letter issued by Student Disability Services (SDS). Please request accommodations for this course as early as possible to provide time for effective communication and arrangements.

For further information or to start the process of requesting accommodations, please contact Student Disability Services at Engineering for Professionals, ep-disability-svcs@jhu.edu.

Student Conduct Code

The fundamental purpose of the JHU regulation of student conduct is to promote and to protect the health, safety, welfare, property, and rights of all members of the University community as well as to promote the orderly operation of the University and to safeguard its property and facilities. As members of the University community, students accept certain responsibilities which support the educational mission and create an environment in which all students are afforded the same opportunity to succeed academically.

For a full description of the code please visit the following website: https://studentaffairs.jhu.edu/policies-guidelines/student-code/

Classroom Climate

JHU is committed to creating a classroom environment that values the diversity of experiences and perspectives that all students bring. Everyone has the right to be treated with dignity and respect. Fostering an inclusive climate is important. Research and experience show that students who interact with peers who are different from themselves learn new things and experience tangible educational outcomes. At no time in this learning process should someone be singled out or treated unequally on the basis of any seen or unseen part of their identity.

If you have concerns in this course about harassment, discrimination, or any unequal treatment, or if you seek accommodations or resources, please reach out to the course instructor directly. Reporting will never impact your course grade. You may also share concerns with your program chair, the Assistant Dean for Diversity and Inclusion, or the Office of Institutional Equity. In handling reports, people will protect your privacy as much as possible, but faculty and staff are required to officially report information for some cases (e.g. sexual harassment).

Course Schedule

Course Schedule :

Module	Date	Module Title	Assignments
1	08/30/21	Introduction	Readings as listed, Lectures and Content , Leadership Discussion Sign-up
2	09/13/21	Factory Method/Strategy/Decorator	Readings as listed , Lectures and Content, Module 2 Assignment
3	09/20/21	Composite/Iterator/Refactoring	Readings as listed , Lectures and Content, Module 3 Assignment, Discussion Leadership
4	09/27/21	Template Method/Abstract Factory/Builder	Readings as listed , Lectures and Content, Module 4 Assignment, Discussion Leadership
5	10/04/21	Singleton and Monostate/Proxy	Readings as listed , Lectures and Content, Module 5 Assignment, Discussion Leadership
6	10/11/21	Idoms and Pattern Languages	Readings as listed , Lectures and Content, Module 6 Assignment, Final Project Topic Submission
7	10/18/21	Adapter/Bridge	Readings as listed , Lectures and Content, Module 7 Assignment, Discussion Leadership
8	10/25/21	Mediator/Observer	Readings as listed , Lectures and Content, Module 8 Assignment, Discussion Leadership
9	11/01/21	Chain of Responsibility/Memento	Readings as listed , Lectures and Content, Module 9 Assignment, Discussion Leadership
10	11/08/21	Analysis Patterns, Architectural Patterns and Antipatterns	Readings as listed , Lectures and Content, Module 10 Assignment, Draft of Final Project Submission
11	11/15/21	Prototype/Command	Readings as listed , Lectures and Content, Module 11 Assignment, Discussion Leadership
12	11/29/21	State/Visitor	Readings as listed , Lectures and Content, Module 12 Assignment, Discussion Leadership
13	12/06/21	Flyweight/Interpreter/Façade	Readings as listed , Lectures and Content, Module 13 Assignment, Discussion Leadership
14	12/13/21	Championing Patterns and Final Project	Lectures and Content, Final Project Submission