SE 310 – Software Architecture I - Fall 2023

Instructors:

Prof. Jeff Salvage  
E-mail: jks29@drexel.edu  
WWW: http://www.cs.drexel.edu/~jsalvage  
Lecture: Section 002: Monday 12:00 PM - 2:50 PM - 3675 Market St. – Room: 910-911  
Office: 3675 Market St. – Room:1162  
Office Hours: Mondays 11:00 AM – 11:50 PM

Prof. Adelaida Medlock  
E-mail: aalban@drexel.edu  
WWW: https://www.cs.drexel.edu/~aalban/  
Lecture: Section 001: Monday: 9:00AM - 11:50 PM - 3675 Market – Room: 910-911  
Office: 3675 Market St. - Room 1064  
Office Hours: Tuesday 3:00 PM – 5:00 PM

Prof. Reza Moradinezhad  
E-mail: rm976@drexel.edu  
WWW:  
Lecture: Section 003: Friday: 9:00AM - 11:50 PM - 3675 Market – Room: 1052  
Office: 3675 Market St. - Room TBD  
Office Hours: TBD

Teaching Assistants:

Fawzia Khan  
ffk23@drexel.edu  
Office hours: Wednesday 12:00 PM - 2:00 PM

Raunaq Malhotra  
rm3552@drexel.edu  
Office hours: Thursday 6:00 PM - 8:00 PM

Maria Mares  
mm4998@drexel.edu  
Office hours: Friday 12:00 PM - 2:00 PM

Office Location CLC: 3675 Market St - Room 1066

Content

Because the Monday sections lose 2 lectures, the Monday classes will follow the 1st schedule. The Friday section will follow the 2nd schedule.

First Schedule – Monday Sections
Week 1 (Monday September 25th)
Synchronous Activities
- No Class, Term starts on a Tuesday.

Asynchronous Activities
- **Video Lecture**: JAVA and OOP Concepts Review (all nine recordings) as needed.
- **Document**: Homework and Lab Exercise Submission Requirements (Read it)
- **Document**: Getting Started with IntelliJ (Read it before class)
- **Student Repository**: Download & review Code and UMLs for JAVA
- **Review Quiz**: Java Code (Take Quiz until scoring >= 70). **Due Monday October 2nd by 05:00 AM EST**.
- **Review Quiz**: Java Concepts (Take Quiz until scoring >= 70) - **Due Monday October 2nd by 05:00 AM EST**.

Week 2 (Monday, October 2nd)
Synchronous Activities
- **Live Lecture**: Course Overview
  - Bring Laptop to class with IntelliJ, Java & StarUML installed.
- **Lab**: Alarm-Radio-Clock. You must attend lab to get credit. Start in class, finish at home. This is individual work. **Due Monday October 9th by 05:00 AM EST**.

Asynchronous Activities
- **Video Lecture**: Chapter 1: Object Oriented Paradigm
- **Video Lecture**: Chapter 3: A Problem That Cries Out for Flexible Code
- **Video Lecture**: Chapter 4: A Standard Object-Oriented Solution
- **Video Lecture**: Chapter 5: An Introduction to Design Patterns
- **Review Quiz**: Chapters 1 - 3 (Take Quiz until scoring >= 70) - **Due Monday October 9th by 05:00 AM EST**.

Week 3 (Monday, October 9th)
Synchronous Activities
- No class, Indigenous People’s Day

Asynchronous Activities
- **Video Lecture**: Chapter 6: The Façade Pattern
- **Video Lecture**: Chapter 7: The Adapter Pattern
- **Video Lecture**: Chapter 8: Expanding our Horizons
- **Video Lecture**: Chapter 9: The Strategy Pattern
- **Video Lecture**: Second Homework Assignment 4 Parts (watch 2 recordings)
- **Review Quiz**: Chapters 4 – 7- **Due Monday October 16th by 05:00 AM EST**.

Week 4 (Monday, October 16th)
Synchronous Activities
- **Homework 1**: Basic Maze and File Parsing – Start in class, finish at home. This is individual work. You must attend lecture. – **Due Monday October 23rd by 05:00 AM EST**.

Asynchronous Activities
- **Video Lecture**: Chapter 10: The Bridge Pattern
• **Pre-Homework Exercise:** 2 slide decks to review and screen capture success. Due Monday October 23rd by 05:00 AM EST.

• **Review Quiz:** Chapters 8 - 10 - Due Monday October 23rd by 05:00 AM EST.

**Week 5 (Monday, October 23rd)**

**Synchronous Activities**

- **Lab: Soda Dilemma - Improving a Bad Design** – Must attend lab and complete assignment during class. This is group work.
- **Lab: Paragraph Processor - Improving a Bad Design** – Must attend lab and complete assignment during class. This is group work.

**Asynchronous Activities**

- **Homework 2 Part A** – Assigned - Due Monday October 30th by 05:00 AM EST.

**Week 6 (Monday, October 30th)**

**Synchronous Activities**

- **Lab: Vehicle Messaging:** Must attend lab and complete assignment during class. This is group work.

**Asynchronous Activities**

- **Video Lecture:** The Factory Patterns
- **Homework 2 Part B:** Assigned - Due Thursday November 9th by 05:00 AM EST.

**Week 7 (Monday, November 6th)**

**Synchronous Activities**

- **Midterm Exam:** During class time

**Asynchronous Activities**

- **Video Lecture:** Chapter 13: Solving the CAD/CAM Problem with Patterns
- **Video Lecture:** Chapter 14: The Principles and Strategies of Design Patterns
- **Video Lecture:** Chapter 15: Commonality and Variability Analysis
- **Video Lecture:** Chapter 17: The Decorator Pattern
- **Video Lecture:** Chapter 18: The Observer Pattern
- **Video Lecture:** Chapter 21: The Singleton Pattern
- **Video Lecture:** The Decorator Revisited
- **Review Quiz:** Chapters 11 - 21 - Due Monday November 13th by 05:00 AM EST.

**Week 8: (Monday, November 13th)**

**Synchronous Activities**

- **Lab:** Advanced Maze - Start in synchronous lecture, finish at home. This is individual work. You must attend lecture - Due Monday November 20th by 05:00 AM EST.

**Asynchronous Activities**

- **Homework 2 Part C:** Assigned - Due Monday November 27th by 05:00 AM EST.
- **Pre-Homework Exercise:** 4 slide decks to review and screen capture success. Due Monday November 20th by 05:00 AM EST.

**Week 9: (Monday, November 20th)**
Synchronous Activities
- **Lab**: Improving a Bad Design (The Observer Pattern) - Must attend lab and complete assignment during class. This is group work.

Asynchronous Activities
- **Homework 2 Part D**: Assigned - *Due Thursday December 4th by 05:00 AM EST.*
- **Video Lecture**: The Command Pattern
- **Video Lecture**: The Template Pattern
- **Video Lecture**: The Builder Pattern
- **Video Lecture**: Iterators
- **Review Quiz**: Template, Command, Builder Patterns - *Due Monday November 27th by 05:00 AM EST.*

**Week 10: (Monday, November 27th)**

Synchronous Activities
- **Lab**: Improving a bad design (The Decorator Pattern) - Must attend lab and complete assignment during class. This is group work.

Asynchronous Activities
- **Video Lecture**: Putting it All Together

**Week 11: (Monday, December 4th)**

Synchronous Activities
- **Review for Exam**

Asynchronous Activities
- None this week

**Week 12: (Monday, December 11th)**

**Final Exam**: Date/Time/Room TBA

---

**Second Schedule – Friday Section**

**Week 1 (Monday September 25th)**

Synchronous Activities
- **Live Lecture**: Course Overview
  - Bring Laptop to class with IntelliJ, Java & StarUML installed.
- **Lab**: *Alarm-Radio-Clock*. You must attend lab to get credit. Start in class, finish at home. This is individual work. *Due Monday October 9th by 05:00 AM EST.*

Asynchronous Activities
- **Video Lecture**: JAVA and OOP Concepts Review (all nine recordings) as needed.
- **Document**: Homework and Lab Exercise Submission Requirements (Read it)
- **Document**: Getting Started with IntelliJ (Read it before class)
- **Student Repository**: Download & review Code and UMLs for JAVA
- **Review Quiz**: Java Code (Take Quiz until scoring >= 70). *Due Monday October 2nd by 05:00 AM EST.*
- **Review Quiz**: Java Concepts (Take Quiz until scoring >= 70) - *Due Monday October 2nd by 05:00 AM EST.*
Week 2 (Monday, October 2nd)
Synchronous Activities
• **Homework 1:** Basic Maze and File Parsing – Start in class, finish at home. This is individual work. You must attend lecture. – Due Monday October 23rd by 05:00 AM EST.

Asynchronous Activities
• **Video Lecture:** Chapter 1: Object Oriented Paradigm
• **Video Lecture:** Chapter 3: A Problem That Cries Out for Flexible Code
• **Video Lecture:** Chapter 4: A Standard Object-Oriented Solution
• **Video Lecture:** Chapter 5: An Introduction to Design Patterns
• **Review Quiz:** Chapters 1 - 3 (Take Quiz until scoring >= 70) - Due Monday October 9th by 05:00 AM EST.

Week 3 (Monday, October 9th)
Synchronous Activities
• **Lab:** Soda Dilemma - **Improving a Bad Design** – Must attend lab and complete assignment during class. This is group work.
• **Lab:** Paragraph Processor - **Improving a Bad Design** – Must attend lab and complete assignment during class. This is group work.

Asynchronous Activities
• **Video Lecture:** Chapter 6: The Façade Pattern
• **Video Lecture:** Chapter 7: The Adapter Pattern
• **Video Lecture:** Chapter 8: Expanding our Horizons
• **Video Lecture:** Chapter 9: The Strategy Pattern
• **Video Lecture:** Second Homework Assignment 4 Parts (watch 2 recordings)
• **Review Quiz:** Chapters 4 – 7 - Due Monday October 16th by 05:00 AM EST.

Week 4 (Monday, October 16th)
Synchronous Activities
• **Lab:** Vehicle Messaging: Must attend lab and complete assignment during class. This is group work.

Asynchronous Activities
• **Video Lecture:** Chapter 10: The Bridge Pattern
• **Pre-Homework Exercise:** 2 slide decks to review and screen capture success. Due Monday October 23rd by 05:00 AM EST
• **Review Quiz:** Chapters 8 - 10 - Due Monday October 23rd by 05:00 AM EST.

Week 5 (Monday, October 23rd)
Synchronous Activities
• **Lab:** Advanced Maze - Start in synchronous lecture, finish at home. This is individual work. You must attend lecture - Due Monday November 20th by 05:00 AM EST.

Asynchronous Activities
• **Homework 2 Part A** – Assigned - Due Monday October 30th by 05:00 AM EST.
• **Video Lecture**: The Factory Patterns

**Week 6 (Monday, October 30th)**

**Synchronous Activities**
- **Lab**: Improving a Bad Design (The Observer Pattern) - Must attend lab and complete assignment during class. This is group work.

**Asynchronous Activities**
- **Video Lecture**: Chapter 13: Solving the CAD/CAM Problem with Patterns
- **Video Lecture**: Chapter 14: The Principles and Strategies of Design Patterns
- **Video Lecture**: Chapter 15: Commonality and Variability Analysis
- **Video Lecture**: Chapter 17: The Decorator Pattern
- **Video Lecture**: Chapter 18: The Observer Pattern
- **Video Lecture**: Chapter 21: The Singleton Pattern
- **Video Lecture**: The Decorator Revisited
- **Review Quiz**: Chapters 11 - 21 - Due Monday November 13th by 05:00 AM EST.
- **Homework 2 Part B**: Assigned - Due Thursday November 9th by 05:00 AM EST.

**Week 7 (Monday, November 6th)**

**Synchronous Activities**
- **Midterm Exam**: During class time

**Asynchronous Activities**
- None this week

**Week 8: (Monday, November 13th)**

**Synchronous Activities**
- **Lab**: Improving a bad design (The Decorator Pattern) - Must attend lab and complete assignment during class. This is group work.

**Asynchronous Activities**
- **Homework 2 Part C**: Assigned - Due Monday November 27th by 05:00 AM EST.
- **Pre-Homework Exercise**: 4 slide decks to review and screen capture success. Due Monday November 20th by 05:00 AM EST.

**Week 9: (Monday, November 20th)**

**Synchronous Activities**
- **Thanksgiving, no class**

**Asynchronous Activities**
- **Homework 2 Part D**: Assigned - Due Thursday December 4th by 05:00 AM EST.
- **Video Lecture**: The Command Pattern
- **Video Lecture**: The Template Pattern
- **Video Lecture**: The Builder Pattern
- **Video Lecture**: Iterators
- **Review Quiz**: Template, Command, Builder Patterns - Due Monday November 27th by 05:00 AM EST.
Week 10: (Monday, November 27th)
Synchronous Activities
  • Extra Help for anyone that needs it.

Asynchronous Activities
  • Video Lecture: Putting it All Together

Week 11: (Monday, December 4th)
Synchronous Activities
  • Review for Exam

Asynchronous Activities
  • None this week

Week 12: (Monday, December 11th)
Final Exam: Date/Time/Room TBA

Texts
We expect that you will have access to these books for assignments.

Required:
Design Patterns Explained: Alan Shalloway & James Trott, Addison and Wesley 2nd Edition. The book can be purchase in either paper or electronic format. eBook is available on Amazon.

Recommended:

For some examinations, we may allow you to have a Java reference book to refer to for details on the language. The book need not be a comprehensive guide to class libraries, but it should allow you to write syntactically correct Java.

JAVA and Integrated Development Environment
This course requires Java 11. Submissions, for homework or lab, using later versions of Java will receive a 0 (zero). Download Java 11 here:


The IDE for this class will be the community edition of IntelliJ. You can download a free copy (having already installed Java on your computer) from:

https://www.jetbrains.com/idea/download/.
Objectives of this course
This course is about good object-oriented design and implementation. In the challenging projects that programmers face, the design stage is where the primary difficulties of programming are worked out. Implementation, the task of turning a design into a working program in a programming language, is supposed to be very straightforward given a good design.

Object oriented programming (OOP) allows programmers to write programs that are easier to reuse and easier to extend and maintain over a life cycle of usage. However, this power is achieved at the cost of making the program designer’s task more complicated, due to the richness, subtlety and peculiar limitations of OO classes and class inheritance. A major objective of this course is to bring students to a level of competency in OO design so that they can handle team or individual projects of moderate complexity. Design skills students should acquire in this course include:

- The ability to decide which classes would be appropriate given only a problem specification,
- Designing for ease of future code maintenance,
- Being able to decide upon details of classes from a problem specification,
- Finding well-known software patterns from typical problem specifications,
- Being able to communicate both static and dynamic aspects of designs using the Universal Modeling Language (UML).

Design skills should be applicable to any of the OO languages in vogue (e.g., C++, Eiffel, Java, etc.).

Grades for design projects will be based, among on other things, on the quality of the design and on the written explanation, description, and justification of the design.

Another objective of this course is to make you confident and competent at implementing an OO design of moderate complexity, using coding techniques based upon generally recognized software engineering principles.

A final objective of this course is to make you more self-aware, descriptive, and reflective about your programming and design. By learning about and practicing with design and quality rationales the course will present, you should be able to state, explain, judge, and justify design or implementation decisions in a way that would be generally acceptable among (at least) entry-level computing professionals. It is also a necessary step towards being able to continue to improve your skill in the future without the need for professors or formal classes.

Prerequisites
We expect that all taking this class has had CS260 (Data structures), CS 265 (Advanced Programming Techniques), and SE 181 (Introduction to Software Engineering and Development) or their equivalent. The experience in these courses includes the design and implementation of programs that involve the use of classes, pointers, and dynamic storage allocation. It should also include the experience of designing several CS 172-like or CS260-
like programs requiring familiarity of class creation, basic I/O, and the use of built-in class libraries. The experience in writing these programs is also expected to have given you knowledge of basic coding practices such as: good selection of names, good quality commenting and indentation, separation of classes into separately compiled files, what constitutes appropriate selection of test data, and how to develop a program incrementally (e.g., use of program stubs as placeholders for as-yet unwritten program modules). Experience with recursive programming design and implementation is expected.

Some assignments will assume that you find it routine to implement programs that use data structures such as trees or linked lists. Some assignments will ask that you will be able to design and implement algorithms to solve simple problems on your own, without extensive hints or outside help. This will probably require a bit more mental effort on your part than many assignments from previous courses, in that you will have to devise the solution algorithm yourself based upon your prior experience with programming problem solutions. You can use the first programming assignment as an indication of what you are expected to have started with. If you have questions, please consult with the instructor.

Grades and Grading

Your grade will depend upon a number of different items, cumulatively. The weighting will be roughly as follows:

- Homework Assignments: 40%
- Lab Assignments: 10%
- Midterm exam: 20%
- Final Exam: 30%
- All quizzes must be taken and passed (70% is the passing grade for each quiz), or you will lose 1 point of your final grade for each quiz not taken/passed. Quizzes can be retaken if failed.

The instructor reserves the right to make modest adjustments (5% or 10% for a category) in the weighting used.

Students will receive a reduced or a failing grade for the course if in the instructor's judgment their performance on the midterm and final examinations indicates that they cannot express an adequate understanding of the material in the course. One criterion used for "adequate" is achieving at least a 40% mark on the final exam.

Good quality in-class participation (being prepared to answer questions posed by the instructor during the class) will be taken into account as a compensating factor for a poor exam performance.

Part of your assignments may consist of written answers rather than just programs. Part of your assignment grade will be based on how you express yourself, as well as what you say. Your writing is expected to be of the quality comparable to technical writing done by a well-regarded computing professional.
In order to pass the course, you **must successfully complete** the programming project (Homework 2). Successfully completing the homework 2 means that the program you submit:

- Includes all four parts
- Complies with all the requirements given in the homework instructions (including grading and tabulating).
- Runs without any errors

### Incomplete Grade Policy

A grade of incomplete (INC) may be considered by your course instructor if exceptional circumstances warrant such course of action. In order to be eligible, you must have:

- successfully completed all the labs.
- successfully completed homework assignment 1.
- successfully completed homework assignment 2, parts A, B & C.
- scored a 70 or more on homework 2 parts B & C.
- taken the midterm and final exams.
- have a legitimate reason to request an incomplete grade.

In cases when the instructor judges that an Incomplete Grade (INC) can be granted, a contract will be signed by both, the student, and the course instructor, and submitted to the Registrar's Office. The coursework must be completed within the following two academic terms otherwise the INC grade will become a failing grade (F).

This policy is consistent with the university's policy found at: [https://drexel.edu/provost/policies-calendars/policies/incomplete_grades/](https://drexel.edu/provost/policies-calendars/policies/incomplete_grades/)

### Handing in assignments

Assignments will be made available on Blackboard Learn and will be due as described in each assignment. The programming assignments must be done in Java using IntelliJ. There may also be written work that should be submitted as specified in the assignment. When assignments call for written responses, they should be created using a word processor, written in intelligible, clear, and correct English. Poorly written or sloppy work will be returned ungraded. Programs that do not compile and cannot be tested will receive little or no “partial credit”.

All submissions must be made through Blackboard Learn. **Make sure what you submit is a final submission, not just a step along the way. We will not open multiple submissions on Blackboard.** All coding submissions, labs and homework assignments, need to include the full project directory, any associated resources (jar files, serialized files, README, other notes) *inside* the zip file. Submissions not following these guidelines will be returned ungraded.

### Assignments Late Policy

- Assignments must be submitted via Blackboard Learn by the given deadline in order to receive full credit.
• Assignments submitted 1 hour to 1 week late will receive a 15% penalty.
• Assignments submitted 1 to 2 weeks late will receive an additional 10% penalty.
• Assignments submitted more than 2 weeks late will be subject to an additional 5% penalty for each week.

**Academic Honesty**
The university's Academic Honesty policy is in effect for this course. You can find the policy at [https://drexel.edu/provost/policies-calendars/policies/academic-integrity/](https://drexel.edu/provost/policies-calendars/policies/academic-integrity/)

You must be the sole original author of all assignments and examination solutions in their entirety, unless the instructor explicitly gives you permission to do otherwise in written directions on an assignment or exam. Design and implementation problems should be discussed with the instructor or the teaching assistants only. This means that you should have an implementation plan that leaves enough time for such consultation. As the university's policy explains penalties up to and including receiving a failing grade for the course with no opportunity to withdraw will be given for first time offenses of plagiarism, fabrication, cheating, or other forms of academic dishonesty. Allowing another student to copy from your work (i.e., helping another student to cheat) is also a violation of the policy on academic honesty.

The standards for originality in a program are similar to those of other written works. Programs by different authors show clear and substantial differences as judged by most criteria, including but not limited to choice of variable and procedure names, line spacing and indentation, choice of program structure, choice of algorithms, ordering of modules, style and content of documentation, module design, and ordering and choice of instructions. The original author of an assignment can explain each detail and how they came to create it on their own. Whenever you use references or sources (i.e., the Internet) for “inspiration” in your programs and designs, you must give a complete and proper citation of the source. Copying or close paraphrasing from other sources and claiming it is your own original work is a violation of the academic honesty policy.

It is your responsibility to avoid violating the university's policy. If you are unclear as to what the policy means in a particular situation, ask the instructor for clarification before you hand anything in.

In addition, be clear that if you electronically copy another person’s work and submit it as your own you are in violation of this policy and could be failed for the course on the first occurrence. If you solicit and/or hire someone to do your work, even if the work is not complete, you are in a violation of this policy and could be failed for the course on the first occurrence. Any violation of this policy may also result in the matter being handed over to an external judiciary panel. Do not let this happen to you.

**Students Accommodations**
In compliance with the Americans with Disabilities Act of 1990, Section 504 of the Rehabilitation Act of 1973, and Drexel University's policies and procedures, the University is committed to the non-discrimination of students with disabilities.
Students requesting accommodations due to a disability at Drexel University need to request a current Accommodations Verification Letter (AVL) in the ClockWork database before accommodations can be made. These requests are received by Disability Resources (DR), who then issues the AVL to the appropriate contacts. For additional information, visit the DR website at [https://drexel.edu/disability-resources/support-accommodations/student-family-resources/](https://drexel.edu/disability-resources/support-accommodations/student-family-resources/) or contact DR for more information by phone at 215-895-1401, or by email at disability@drexel.edu

**Other important Academic Policies**

In addition to the course policies listed on this syllabus, the following University policies are in effect in this course:

- **Academic Integrity**: [https://drexel.edu/studentlife/community-standards/code-of-conduct/academic-integrity-policy](https://drexel.edu/studentlife/community-standards/code-of-conduct/academic-integrity-policy)
- **Course Drop Policy**: [http://www.drexel.edu/provost/policies/course-add-drop/](http://www.drexel.edu/provost/policies/course-add-drop/)
- **Course Withdrawal Policy**: [https://drexel.edu/provost/policies-calendars/policies/course-coop-withdrawal/](https://drexel.edu/provost/policies-calendars/policies/course-coop-withdrawal/)
- **Absence from class due to University sponsored activities**: [https://drexel.edu/provost/policies-calendars/policies/absence/](https://drexel.edu/provost/policies-calendars/policies/absence/)
- **Student athletes who will miss class due to an athletic event (game, tournament trip, etc.) must provide the course instructor with an official letter from the Drexel Department of Athletics that outlines the dates and times of their absence due to the sporting event(s).**
- **Drexel Student Learning Priorities**: [https://drexel.edu/institutionalresearch/assessment/outcomes/dslp/](https://drexel.edu/institutionalresearch/assessment/outcomes/dslp/)

**Note on Attendance**

Under Title IV (“Student Assistance”) of the Higher Education Act of 1965, institutions of higher education are required to verify that students receiving financial aid have begun Initial Course Participation (ICP). ICP refers to the student’s attendance in class during the term for which they are receiving financial aid. Class attendance is critical to your success as a student. Missing classes may impact your class success and your federal financial aid.