CS 519 Syllabus

Semester:	Spring 2025
Course title:	Mathematics of Digital Signal Processing II
Instructor:	Professor Matt Klassen
Email:	mklassen@digipen.edu
Phone:	(425) 895-4423
Office hours:	T,W 2:00-3:00, or by appointment
Course Web Page:	http://azrael.digipen.edu/MAT321
Time/Place:	M,F 2:00-3:20 in Pascal

WEB PAGES AND MOODLE:

The Moodle page for CS 519 is the same as MAT 321 and will contain a link to the course web page. The web page is the central repository for all course documents, including homework assignments. Updates to homework will be posted on the web page only. Scores for quizzes, homework, exams, and projects, will be posted through perl scripts on the course web page and also on Moodle.

The Moodle page will be used for chat, forums, and for submission of homework, projects and quiz and exam scratch work.

The Teams channel may be used for lectures. Students may record lectures and review them in Teams. Recording with Teams is convenient but not high quality. For best quality, recording with OBS is recommended. Please follow DigiPen guidelines for requesting streaming of lectures.

MATERIALS:

Text: *A Digital Signal Processing Primer*, by Ken Steiglitz. The course is based on this text, lecture material, notes, and homework.

Reference Materials (not required) :

Digital Signal Processing, by Oppenhiem and Schafer
Digital Filters for Everyone, by Rusty Allred
Music: A Mathematical Offering, by David Benson
Schaum's Outline: Digital Signal Processing, by Monson Hayes
On the Sensations of Tone, by Hermann Helmholtz
Real Sound Synthesis for Interactive Applications, by Perry Cook
Signals, Sound, and Sensation, by William Hartmann
Designing Sound, by Andy Farnell

COURSE DESCRIPTION:

This course continues to explore the mathematical foundations of digital signal processing, with applications to digital audio programming. Topics include: Review of digital signals, Z-transforms and convolution, filter types, applications of fast Fourier transform, switching signals on and off, windowing, spectrograms, aliasing, digital to analog conversion, Nyquist Theorem, filter design, Butterworth filters, reverb, and the phase vocoder.

COURSE GOALS AND OBJECTIVES:

1) Students will learn some of the basic algorithms in digital signal processing using complex numbers, such as the Discrete and Fast Fourier Transforms, and various filters.

2) Students will become familiar with basic concepts of digital signal processing and will describe these from the mathematical perspective.

3) Students will solidify their knowledge of Linear Algebra and Calculus by using these subjects as tools to solve problems involving vector spaces of complex numbers, continuous functions, and discrete functions and signals.

QUIZZES AND EXAMS:

Quizzes will be given periodically to test comprehension of lecture material. There are no make up quizzes, but I do drop your lowest two quiz scores. The quizzes will last for approximately twenty minutes. Quizzes are posted online on the course website during class. Quizzes and exams can be done online. Submission of answers is through the website and submission of scratchwork is through Moodle. Both submissions are required by the due date/time to avoid a score of zero.

For multiplie choice quizzes and exams, please follow these procedures: Work out the quiz problems and circle your answers on the question sheet. When you are finished, transfer the answers to the answer sheet. Go to a web browser and enter the answers online. Under no circumstances are you allowed to discuss the quiz questions with any other student during the quiz or the data entry process. Scratch work may be used to verify that you did your own work. Your scores will be posted online by your student ID.

There will be a midterm exam given during regular class hours, and a final exam. There are *NO* make up exams unless you have a *compelling and well documented reason* for missing a test.

Calculators, notes, books, are allowed on quizzes and exams.

GRADING:

Midterm Exam	20%
Final Exam	20%
Homework	20%
Quiz	20%
Projects	20%

Grades will be determined based on total course percentage. Percentage scores will determine letter grades according to the scale: (in the worst case)

A	$93-100$	A-	$90-92.9$		
B+	$87-89.9$	B	$83-86.9$	B-	$80-82.9$
C+	$77-79.9$	C	$73-76.9$	C-	$70-72.9$
D	$60-69.9$	F	<60		

ACADEMIC INTEGRITY:

Academic dishonesty, or cheating, occurs when a student represents someone else's work as their own, or assists another student in doing so. This can happen on exams, quizzes, homework, or projects. Academic dishonesty also may occur when a student uses any prohibited reference or equipment in the completion of a task. For example, the use of a calculator, notes, books or the internet when it is prohibited. Plagiarism is a common form of academic dishonesty. This can take the form of copying and pasting exerpts from the web, and representing them as original work. The type and severity of any occurence, as well as the legitimacy of any claim of academic dishonesty, will be judged by the instructor and the disciplinary committee. All students are asked to help in promoting a culture of academic integrity by discouraging cheating in all forms.

HOMEWORK ASSIGNMENTS:

Homework will be assigned and posted on the web page and collected weekly. You are responsible for checking the web page and noting the assignments and the due date. You may work on homework together, as well as consult the tutors and the instructor. However, the final work that you turn in must be your own work.

PROJECTS:

The full description of the programming project can be found on the course web page. Submission of projects should be in a zipped folder which contains source and executable and which can be uploaded on the Moodle page.

COMPUTATIONAL RESOURCES:

You are encouraged to do linear algebra and other calculations for the homework using a calculator or symbolic package such as SciLab. The symbolic algebra package SciLab is free and open-source, and is similar to MATLAB.

DISABLED STUDENT SERVICES:

Students with physical, psychological or learning disabilities that affect their ability to perform major life activities associated with this class may be eligible for reasonable accommodations under the Americans with Disabilities Act. If you have a documented disability please contact the Disability Support Services office to arrange for accommodations for this class.

GRADUATE STUDENTS:

Graduate students will be required to do a class presentation or report which will be factored in as part of their homework grade. The subject matter will involve an in depth analysis of a particular problem, or an application, and will require some synthesis of current work and contributions in the field. This will form part of the graduate student exposure to research methods in course work.

TENTATIVE WEEKLY TOPICS:

Week	Dates	Topics
1	Jan 6 - 10	time/frequency domains, discrete-time phasors
2	Jan 13 - 17	Z-transform and convolution
3	Jan 21 - 24	inverse Z-transform, stability
4	Jan 27 - Jan 31	Applications of the FFT, switching signals on and off
5	Feb 3 - Feb 6	Windowing, spectrograms
6	Feb 10 - 14	Periodization and the continuous Fourier transform
7	Feb 18 - 21	Fourier transform of rect, sinc, and Gaussian, Midterm Exam F Feb 21
8	Feb 24 - Feb 28	Convolution integral for continuous functions
9	Mar 3 - Mar 7	Schwartz functions and Distributions as limits
10	Mar 10 - 14	Shift and derivative properties for Fourier transform
	Mar 17 - 21	Spring Break
11	Mar 24 - 28	Sampling and Interpolation: from continuous to discrete
12	Mar 31 - Apr 4	Nyquist theorem, digital to analog conversion
13	Apr 7 - 11	reverb, oversampling
14	Apr 14 - 18	filter design and implementation
15	Apr 21 - 25	Final Exams

GENERATIVE AI POLICY:

Students are encouraged to experiment with generative AI to assist in both homework and coding. The primary viewpoint should be the following: all such activity is for the purpose of discussion, just as with a human being. If the AI gives a solution to a problem, you should suspect that it might be incorrect, just as you would with a human. You should never copy verbatim, any written work or code, just as with a human collaborator. Your work that you turn in must be done by you, and you must be confident that it is correct based on your own verification. If any significant portion of your work or code was based on suggestions made by an AI, then you should document this. This must include: excerpts of the prompt you used, and the response of the AI that you found helpful if in fact this led you to use this response to develop your answer or your code. Any prompt/response that is not useful in modifying your work in a significant way is not necessary to report. For example, if you use an AI to look at a string of calculations that you think might have an error in it, and the AI finds an error in a simple calculation, then you do not need to report it. Also, at the other extreme, if you have a discussion with an AI that helps to clarify your approach to a problem, and then you go and do the problem with this clarification, you do not need to report it. Again, this would be like consulting with another student, who helped to get you on the right track but then you did the work on your own. Students who copy the work of an AI will be considered plagiarizing. This type of activity can often be caught due to strange errors in written work or code. Never copy. For a code example, suppose you asked the AI how to write a function in C++ with specific inputs and outputs. You look at the code and suddenly you see the idea that you were missing. So you can go and write the function yourself. In this case you should document how the AI helped you, your prompt and the response. It will then also be clear that you have rewritten the code yourself. If you can verify that the function the AI wrote works, by testing it, but you dont understand how it works in detail, then you should stop and think of ways to gain a better understanding. There are many ways, the simplest being to ask more questions with careful prompts, to gain a better understanding. This is a good way to learn, and again it should also be documented, as long as it is helpful and it has affected your final solution to a problem in a significant way.