Course Syllabus

Jump to Today ✎ Edit

Fundamentals of Music Processing: Fall 2024

Meets: TR 1-2:30, 4-270

See **Course Overview (https://canvas.mit.edu/courses/27586/pages/course-overview)** for a description of the course, office hours, and class expectations.

Schedule

Week of	Mon	Tue	Wed	Thu	Fri
Sep 1				0: Intro / Class Overview	
Sep 8	setup due	1: Music Representation		Lab 1	
Sep 15		2: Kick / Snare Classifier	pset 1 due	Lab 2	
Sep 22		3a: Fourier Transform 1	pset 2 due	3b: Fourier Transform 2	
Sep 29		Lab 3		4a: Chromagrams	
Oct 6		4b: Music Synchronization	pset 3 due	Lab 4	
Oct 13		*IPD (no class)*		5a: Onsets, Tempograms, Pulse Estimation	
Oct 20	pset 4 due	5b: Beat Detection		Lab 5	
Oct 27		6a: Chord Recognition	pset 5 due	6b: Chord Recognition w/HMMs	
Nov 3		Lab 6		7: Structure Analysis	FP topic
Nov 10		Lab 7	pset 6 due	8: Audio Retrieval	FP proposal
Nov 17	pset 7 due	Lab 8		9: Audio Decomposition	
Nov 24	pset 8 due	FP: Milestone review (grad students only)		*Thanksgiving (no class)*	
Dec 1		Lab 9		FP: Project Presentations (everyone attends)	
Dec 8	pset 9 due	(no class)			

Color code:

- Assignment Due
- Final Project (grad only)

Course Overview

21M.387 / 21M.587 / 6.3020 Fundamentals of Music Processing

- Term: Fall 2024
- Meets: TR 1:00-2:30 in 4-270
- Instructor: **Eran Egozy** (https://canvas.mit.edu/courses/27586/pages/about-your-instructors) (**egozy@mit.edu** (mailto:egozy@mit.edu))
  - Office hours: Tuesday 3-4pm & Wednesday 2-3pm in W18-2313, or **on zoom** (https://mit.zoom.us/my/egozy) by appointment (just email me)
- TA: **Alyssa Solomon** (https://canvas.mit.edu/courses/27586/pages/about-your-instructors) (**solam@mit.edu** (mailto:solam@mit.edu))
  - Office hours: Monday 1-2pm in 36-155 and Thursday 3-4pm in W18-2310 or by appt (feel free to email).

See the **Syllabus** (https://canvas.mit.edu/courses/27586/assignments/syllabus) for the course schedule and deadlines.

Description

Fundamentals of Music Processing deals with music analysis in the audio domain as a signal-processing problem. In the same way that speech processing uses signal analysis to understand spoken words, music processing uses signal analysis on music waveforms to understand higher level musical structure.

We begin by introducing Fourier analysis using the Discrete Fourier Transform, and its commonly used relative, the Short-Time Fourier Transform. From there we study features extraction methods such as chroma analysis and onset detection, and examine analysis tools like dynamic-time-warping and self-similarity matrices. These techniques serve to develop a variety of music analysis applications, including:

- Onset classification
- Temporal alignment of different renditions of the same music
- Automatic chord recognition and key detection
- Structural analysis of music
- Tempo and beat tracking
- Content-based audio retrieval
- Music-based audio decomposition

Pre-requisites: 21M.051, 6.3000 (was 6.003), and python proficiency.

Text and Software

The required textbook for this class is **Fundamentals of Music Processing (2nd edition)** (https://www.springer.com/gp/book/9783030698072#aboutBook) by Meinard Müller. You may purchase a copy or **download a PDF with your MIT credentials.** (https://mit.primo.exlibrisgroup.com/discovery/fulldisplay?

docid=alma9935156294006761&vid=01MIT_INST:MIT) An additional optional but very useful text is ***Mathematics of the Discrete Fourier Transform*** *(https://www.amazon.com/Mathematics-Discrete-Fourier-Transform-DFT/dp/097456074X)* by Julius O Smith III. It is available for **online viewing here** (https://ccrma.stanford.edu/~jos/mdft/mdft.html) .

This class is taught using Jupyter Notebook and python. All lectures, labs, and psets are structured as notebook (.ipynb) files. Installation Instructions are **provided here** (https://canvas.mit.edu/courses/27586/assignments/346809) .

Class Structure and Assignments

The course is organized into nine units of study. Each unit consists of the following:

Reading assignments are due before the first unit's lecture. Please familiarize yourself with the topics. It will make the in-class material more understandable.

Lectures consist of conceptual presentations and live demonstrations in Jupyter Notebooks. Some units have a single lecture while others have two, depending on the depth of the particular unit. You will have access to all lecture notebooks so that you can review them on your own.

Labs consist of in-class exercises that reinforce lecture materials and coding skills. They also prepare you for the unit's homework assignment. You will work on labs during class time with randomly assigned partners. Each lab needs to be checked off by the instructor or TA at the end of the class. If you did not get a checkoff, you must upload your completed lab by end-of-day to receive credit.

Problem sets are due by 11:59pm on Mondays or Wednesdays. Please check the **Syllabus** (https://canvas.mit.edu/courses/27586/assignments/syllabus) or **Assignments** (https://canvas.mit.edu/courses/27586/assignments) page for each assignment's deadline. You may work on assignments alone or with someone from class. Try **pset partners** (https://psetpartners.mit.edu/) if you are looking for a buddy. If you work together, you can discuss ideas, but each person must write their own code. You must also accompany your pset submission with a note describing how you worked together. Psets should be uploaded to Canvas.

Complete the **Final Project** (https://canvas.mit.edu/courses/27586/pages/final-project) if you are taking the graduate version of this class (21M.587). Most of this work will start in the 2nd half of the semester. You will present this project on the last day of class.

Attendance and Support

While you will certainly learn from the text and lecture notebooks, attending the live lectures and working in lab are invaluable towards mastering the material. Attendance, therefore, is mandatory and is a part of your grade.

That said, stuff happens. If you need to miss class due to an extenuating circumstance, email us (instructor and TA) in advance and you may be excused in part or in full on a case-by-case basis. You are granted accommodation for missing class due to religious / faith practice, but please let us know in advance. If you don't feel comfortable coming to one of us, you may **contact S3 for an excuse note** (https://studentlife.mit.edu/s3/support-advocacy/excuse-notes) .

We will endeavor to record and post lecture videos to help you catch up if you miss a class. You should also contact classmates to find out what you missed.

And as always, please contact S3 or graduate support for any issues (medical, personal, or otherwise) that hinder your ability to focus on academics. You can reach out to a dean you have worked with in the past, join their **virtual help queue** (https://sicp-s3.mit.edu/queue) , or e-mail **s3-support@mit.edu** (mailto:s3-support@mit.edu.) .

Grading

Grading varies slightly depending on if you are enrolled in 21M.387 (undergrad version) or 21M.587 (graduate version).

	Undergraduate	**Graduate**
Attendance	10%	5%
Labs	15%	10%
Psets	75%	70%
Final Project	N/A	15%

Grade cutoffs are the standard set with [97, 100] = A+, [93, 97) = A, [90, 93) = A-, [87, 90) = B+, etc...

There is no final exam.

Deadlines and Extensions

Late psets are penalized 10/24% per hour late (i.e., 10% per day), with a maximum penalty of 50%.

Every student is allowed two pset extensions, which must be requested before the deadline using the link on the course **Home Page** (https://canvas.mit.edu/courses/27586/pages/home-page) . When you request an extension, you must propose a new, reasonable deadline. A reasonable deadline is one that you feel confident hitting given your overall workload, but that won't put you too far behind in the class. If you need an additional extension, you must first get approval from a dean (**S3** (mailto:s3-support@mit.edu.) or **graduate support** (mailto:gradsupport@mit.edu)) and then email the instructor to make your request.

Academic Integrity and Collaboration

Read MIT's **Academic Integrity Policy** (http://integrity.mit.edu/) . It applies here. In particular, you will be coding for this class, so read the **Writing Code** (http://integrity.mit.edu/handbook/writing-code) section.

If working as a team, each team member should contribute an equal share of the work for the assignment. You can discuss ideas and approaches together, but each person must write and submit their own code. Every collaborative assignment submission must include a note describing the nature of the collaboration: who did what, and how the collaboration worked.

You may look online for python implementation tips and ideas when doing assignments. Stack Overflow is a great resource, for example. Cite the source as a code comment. If you do copy, you should limit yourself to a few lines of code. Do not copy or use whole modules or large functions.

Do not use ChatGPT, GitHub Copilot, or any other code generators. This is considered cheating and often produces incorrect answers anyway.

The units' notebooks make extensive use of `fmplib`, a python module we wrote specifically for this class. Sometimes, a function that you need to write for a pset is already implemented in `fmplib`. You are allowed to test the results of your function against the `fmplib` version, but you are not allowed to check the source code of `fmplib`.

Please do not post any code we distribute to you online. If, for example, you want to use GitHub to manage your course materials, make sure to do so in a private repo.

Disability and Access Services

MIT is committed to the principle of equal access. Students who need disability accommodations are encouraged to speak with Disability and Access Services (DAS), prior to or early in the semester so that accommodation requests can be evaluated and addressed in a timely fashion. If you have a disability and are not planning to use accommodations, it is still recommended that you meet with DAS staff to familiarize yourself with their services and resources. Please visit the **DAS website (https://studentlife.mit.edu/das)** for contact information.

If you are approved for accommodations, we're ready to assist with implementation - just let us know.