New Directions in Computing:

Program Synthesis

COMS 3997: Spring 2023

Instructor Info

Mark Santolucito

Office Hrs: TBA

505 Milstein

msantolu@barnard.edu

Course Info —

Prereq: PLT

Thur

170 min Lecture, 1x/week

Virtual

Capped at 24 students

Overview

This course will give an introduction to program synthesis, the process of writing programs that write programs. Program synthesis is an area of study at the intersection of programming languages, formal methods and AI. In this course we will focus on the programming languages and formal methods aspects of program synthesis. We will cover three types of program synthesis: 1) inductive synthesis (programming-by-example) 2) functional synthesis (synthesis from first order logic formulas) 3) reactive synthesis (synthesis from temporal logic formulas). For each unit we will ask three key questions: 1) What is the language interface used to specify the desired behavior of a program 2) How do we construct/find a program that both meets the specified behavior and is the program the user expected 3) How do we assist the user in writing a specification for synthesis This course will involve reading and discussing academic papers, using state-of-the-art program synthesis tools, and writing basic program synthesizers. These topics will require advanced logical reasoning/mathematics, as well as a solid understanding of programming language foundations. The course will feel something like a cross between discrete math and PLT.

Grading Scheme

10%	Participation
10%	Homework 1
15%	Homework 2
10%	Homework 3
15%	Homework 4
10%	Homework 5
10%	In class presentation
20%	Final Project

Learning Outcomes

- Ability to organize and manage software development infrastructure
- An understanding of the tools and technologies used in the audio industry for plugin development
- A conceptual framework for the thought, design, and development processes that go into the creation of commercial software products.
- Hands-on experience with state-of-the-art software engineering techniques

Material

Required Text (available through Columbia Library)

Designing Audio Effect Plugins in C++ For AAX, AU, and VST3 With DSP Theory, Will C. Pirkle, 2019

Recommended Text

Intro to Audio Plugin Development, Jacob Penn, Output https://www.kadenze.com/courses/intro-to-audio-plugin-development/info

Late Policy

Any late homework submission assignment will be docked 50% of total possible points on that assignment, up to two weeks after original due date, at which point no credit will be awarded. This policy does not apply to the final project, which cannot be accepted after the due date except in exceptional circumstances.

Class Attendance Policy

Beyond the 10% of your grade that is allocated to participation as stated above, you are expected to attend every class period. Excessive absences will require consultation.

Class Laptop Policy

Some classes will require the use of laptops. If using a laptop in class is not possible for any student, we will pair program. In class meetings that do not explicitly note the need for a laptop, all digital assistants (laptops, phones, smart watches, AR glasses, etc) should be stored for the entirety of class.

FAQs

❓ Is this class easier than PLT?

❗ No, expect to earn a lower grade than you would have earned in PLT.

❓ Does this class consider accesibility?

❗ Not directly, we will have a lot of theory to cover, so issues of accesibility will largely be out of scope.

Honor Code

You are expected to hold yourself to the highest standard of academic integrity and honesty, as reflected in the Barnard Honor Code. Approved by the student body in 1912 and updated in 2016, the Code states:

We, the students of Barnard College, resolve to uphold the honor of the College by engaging with integrity in all of our academic pursuits. We affirm that academic integrity is the honorable creation and presentation of our own work. We acknowledge that it is our responsibility to seek clarification of proper forms of collaboration and use of academic resources in all assignments or exams. We consider academic integrity to include the proper use and care for all print, electronic, or other academic resources. We will respect the rights of others to engage in pursuit of learning in order to uphold our commitment to honor. We pledge to do all that is in our power to create a spirit of honesty and honor for its own sake.

Wellness Statement

It is important for undergraduates to recognize and identify the different pressures, burdens, and stressors you may be facing, whether personal, emotional, physical, financial, mental, or academic. We as a community urge you to make yourself–your own health, sanity, and wellness–your priority throughout this term and your career here. Sleep, exercise, and eating well can all be a part of a healthy regimen to cope with stress. Resources exist to support you in several sectors of your life, and we encourage you to make use of them. Should you have any questions about navigating these resources, please visit these sites:

- `http://barnard.edu/primarycare`
- `https://barnard.edu/about-counseling`
- `http://barnard.edu/wellwoman/about`
- Stressbusters Support Network

Center for Accessibility Resources & Disability Services

If you believe you may encounter barriers to the academic environment due to a documented disability or emerging health challenges, please feel free to contact me and/or the Center for Accessibility Resources & Disability Services (CARDS). Any student with approved academic accommodations is encouraged to contact me during office hours or via email. If you have questions regarding registering a disability or receiving accommodations for the semester, please contact CARDS at (212) 854-4634, cards@barnard.edu, or learn more at `barnard.edu/disabilityservices`. CARDS is located in 101 Altschul Hall.

Affordable Access to Course Texts & Materials

All students deserve to be able to study and make use of course texts and materials regardless of cost. Barnard librarians have partnered with students, faculty, and staff to find ways to increase student access to textbooks. By the first day of advance registration for each term, faculty will have provided information about required texts for each course on CourseWorks (including ISBN or author, title, publisher, copyright date, and price), which can be viewed by students. A number of cost-free or low-cost methods for accessing some types of courses texts are detailed on the Barnard Library Textbook Affordability guide (library.barnard.edu/textbook-affordability). Undergraduate students who identify as first-generation and/or low-income students may check out items from the FLIP lending libraries in the Barnard Library (library.barnard.edu/flip) and in Butler Library for an entire semester. Students may also consult with their professors, the Dean of Studies, and the Financial Aid Office about additional affordable alternatives for having access to course texts. Visit the guide and talk to your professors and your librarian for more details.

Class Schedule

TODO

1. Homework 1 - Write a brute force algorithm to solve a programming by example query. The recommended language is JavaScript, but any language that allows for evaluating strings as code at runtime will work. You should 1) decide on a input format for the user provided examples, 2) enumerate programs (as strings) however you like and 3) use an “‘eval“‘-like function to check that a potential solution matches the given input-output pair.
2. Homework 2 - Use a smarter algorithm for PBE
3. Homework 3 - Something basic with calling a SMT solver
4. Homework 4 - Use either Rosette (more structured option) or SyGuS (more flexible option - more to do yourself, but less documentation to read) to synthesize programs in a language of your choosing
5. Homework 5 - Use tsltools to synthesize programs in a new language/application domain
6. Final Project - In a group, use some collection of the tools we have covered in class to build a new synthesis tool. You will likely want to focus on using existing tools to tackle a new application area, but new approaches to synthesis are also welcome.

MODULE 1: Inductive Synthesis

Week 1 (Jan 19)	Class introduction + The problem of Program Synthesis	
Week 2 (Jan 26)	Inductive Synthesis + Bottom up search	
Week 3 (Feb 2)	Top down/type directed search	Homework 1 due
Week 4 (Feb 9)	Sketch	
Week 5 (Feb 16)	Deductive synthesis and EGGs	Homework 2 due

MODULE 2: Functional Synthesis

Week 6 (Feb 23)	Intro to SAT/SMT solvers	Online class
Week 7 (March 2)	SyGuS	
Week 8 (March 9)	Rosette and Synquid	Homework 3 due
Week X (March 16)	no class, spring break	
Week 9 (March 23)	SemGuS	Homework 4 due

MODULE 3: Reactive Synthesis

Week 10 (March 30)	Intro to Reactive Systems and Temporal Logics	Online class
Week 11 (April 6)	LTL Synthesis Procedures	
Week 12 (April 13)	Beyond Booleans in Temporal Logic	Homework 5
Week 13 (April 20)	Program synthesis + Machine learning (everything we haven’t covered)	

Week 14 (April 27)	Final Project in-progress presentations

Disclaimer

This syllabus, the nature and number of projects, readings, topics, etc, are subject to change either by necessity or design. Any changes will be reflected in a new syllabus and/or announced in class and on the course web platform.