

CS302: Paradigms of Programming

Introduction and Logistics

Manas Thakur



Spring 2021

Natural languages

- Name the languages that you can speak/read/write.
- A unique advantage of living in India?
 - The PoP course in no other country can start with this slide!
- Have you tried learning a new language?
- How different was that language from those you already knew?
- What was your strategy?

Programming languages

- Name the PLs that you know.
 - What do we mean by *know*?
- Name an English letter that's not a PL.
- Why do we have so many PLs?
- How did so many PLs evolve?
- Or aren't they so many, but just *syntactic sugars* over some basic constructs?

The Purpose of Things

“The purpose of life is a life of purpose.” - Robert Byrne

- What’s the purpose of a programming language?
 - To specify programs
- What’s the purpose of a program?
 - To specify computations
- What all is **computable**?

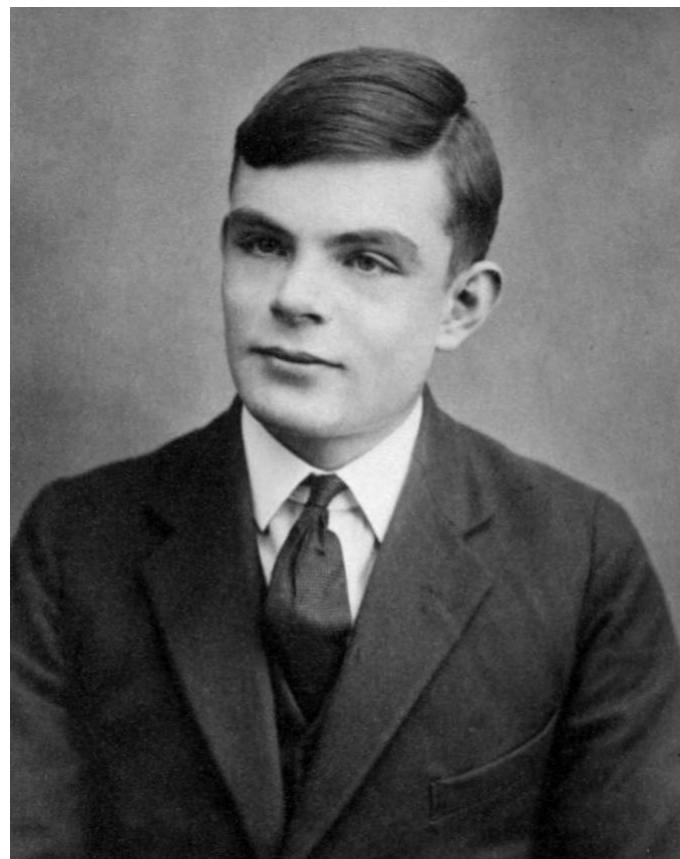
virtual

Look at the board!



Standing on the shoulders of giants

- Who are these people?
- Hints:
 - Both were computer scientists.
 - The most famous award in CS is named after the LHS.
 - RHS was the PhD advisor of the LHS.



Alan Turing



Alonzo Church

Two models of computation

- The **Turing Machine** (1936) performs computations by:

- Reading input

If you don't remember FLAT,
think of a computer that you are used to!!

- Modifying an internal memory using instructions

- Producing output

- The **Lambda Calculus** (1936) performs computations by:

- Evaluating expressions

We will learn more of this in the course.

- **Church-Turing Thesis** (1937): Both the models, and all other *reasonable* computation models, are equivalent!

Paradigms of Programming

- Computation is much more fundamental (and older) than computers or computer science.
- There are several paradigms (aka *ways*) of specifying computations.
- But not as many as there are programming languages.
- Understanding the different paradigms of programming
 - empowers us to **choose** the right PL for the task at hand;
 - makes **learning** a new PL much easier;
 - equips us in **designing** a new PL or its *evaluator*;
 - is huge **fun**!

Some side (main) learnings?

- Standard ways of comprehending/designing large programs
- The crux of future programming languages
- How beauty leads to simplicity
- Different *right* ways of thinking about the same thing
- Motivation to learn the way of cooking (CS502) the food (CS302) that everyone (CSXYZ) eats

Rough sketch of the course

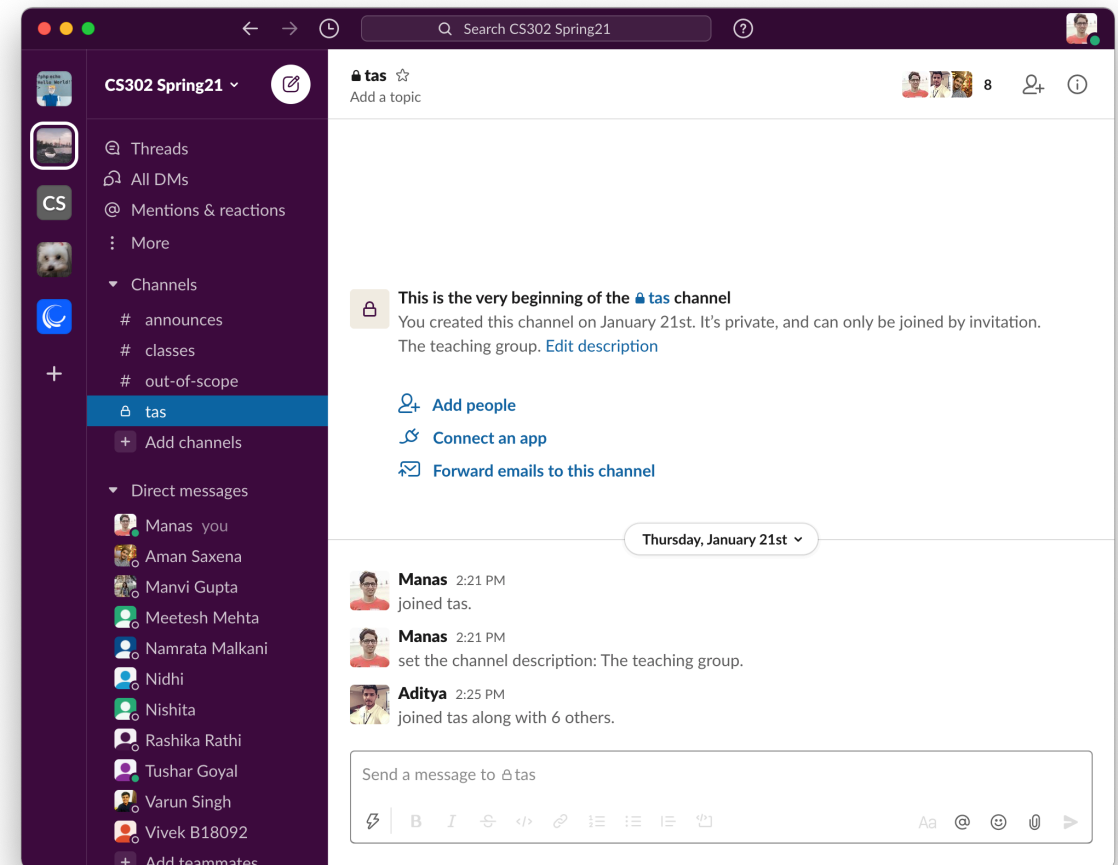
- **Functional** Programming
 - Concepts: Scheme; Extended examples: λ -calculus and Haskell
- **Object-Oriented** and **Imperative** Programming
 - Concepts: Scheme (again!); Extended example: Java
- **Logic** Programming
 - Concepts: Scheme (yes, yet again!); Extended example: Prolog

Logistics

- Three **classes** per week:
 - **C slot:** Monday 10 am; Wednesday 11 am; Friday 8 am or Thursday 2 pm?
- One **lab** every one/two weeks:
 - **L2 slot:** Tuesday 2 pm, starting next week
- **URLs:**
 - Course webpage (schedule, slides/notes, deadlines, textbooks, links):
manas.gitlab.io/courses/pop/spring21/
 - Moodle (submissions, feedback, marks): *You know better!*
 - Slack (doubts, discussions, and much more offline fun):
cs302spring21.slack.com
 - Classes (GMeet); will be uploaded *weekly* on an “unlisted” YouTube playlist

Meet the Team

- Instructor: Manas Thakur
- TAs:
 - Aditya Anand
 - Meetesh Mehta
 - Aman Saxena, Anvay Shah, Manvi Gupta, Namrata Malkani, (Rishi Sharma?), Saurabh Bansal, Varun Singh
- Find everyone on **Slack!**



The Constitution

- **Rights:**

- An in-depth understanding of the various paradigms of programming
- Acceptable answers to doubts/queries
- Timely assignment/quiz evaluations; goal: 10 days from the submission/exam date

- **Duties:**

- Sincerity and honesty
- No hesitation in reaching out to the instructor/TAs
- **No plagiarism** in assignments

No means no:

- “I just saw my friend’s code...”
- “I found it online...”
- “We discussed the assignment and hence obviously our code will have common elements...”

- **Institute norms:**

- Attendance (not compulsory, but recommended 100%)
- Grading (standard relative grading as per institute policy)

Learning resources

Most
concepts and
Scheme

Lambda
Calculus and
Prolog

Some
introduction to
Haskell

- Textbooks (**SICP**, PLCC, LYAH)
- Some videos and papers
- Take notes in class (slides won't always be enough/available)
- Practice programs in the lab and at home

Evaluation

- **Exams (60%):**
 - Quiz 1 (15%)
 - Quiz 2 (15%)
 - End sem (30%)
- **Take-home assignments (30%)**
- **Labs and others* (10%)**

*Hold on

Homeworksss

- Comprehension **C1**: (Remember *others*?)
 - Read first seven pages of John Backus's Turing Award Lecture
 - Try finding out why was that lecture **very** interesting
 - Submit a short summary by Sunday 11:55 pm; get **1 mark**!
- Install **DrRacket** with **sicp** package on your computer
 - <https://download.racket-lang.org/>
- **Next class: Back to high-school!**



Log out to log in

Dedication text of ***Structure and Interpretation of Computer Programs (SICP)***

Authors: **Harold Abelson and Gerald Jay Sussman with Julie Sussman (MIT)**

THIS BOOK IS DEDICATED, in respect and admiration, to the spirit that lives in the computer:

“I think that it’s extraordinarily important that we in computer science keep fun in computing. When it started out, it was an awful lot of fun. Of course, the paying customers got shafted every now and then, and after a while we began to take their complaints seriously. We began to feel as if we really were responsible for the successful, error-free perfect use of these machines. I don’t think we are. I think we’re responsible for stretching them, setting them off in new directions, and keeping fun in the house. I hope the field of computer science never loses its sense of fun. Above all, I hope we don’t become missionaries. Don’t feel as if you’re Bible salesmen. The world has too many of those already. What you know about computing other people will learn. Don’t feel as if the key to successful computing is only in your hands. What’s in your hands, I think and hope, is intelligence: the ability to see the machine as more than when you were first led up to it, that you can make it more.”

— Alan J. Perlis (April 1, 1922 – February 7, 1990)