

CSC 372, Spring 2025

# Comparative Programming Languages

*Michelle Strout*



January 16, 2025

# Course Description

- **Introduction to several major high-level programming languages and their characteristics. Programming projects are required in at least three languages.**
- **Main Goal: Be able to compare programming languages as tools to solve problems**

# Instructor Introduction:

## Dr. Michelle Strout

BS, MS, and Ph.D. at  
University of  
California, San Diego



Full Professor at University of Arizona 2015 thru  
2023, currently Affiliate Professor



Senior Engineering Manager  
2021-2024 at Hewlett Packard  
Enterprise, currently a  
Distinguished Technologist



Kids have gone to UofA



# TA Introductions

- **Phyllis Spence**

- 1<sup>st</sup> year PhD student
- Have a masters in Computer Science
- 30+ years in Software Engineering industry, 26 in Aerospace Industry
- Professionally trained musician, vocalist, classical piano, guitar, ...
- Love gigantic jigsaw puzzles, have done 43K+ pieces one

- **CJ Chen**

- Senior CS minor, Math major, Music minor
- Did an internship last summer, research internship at Stanford
- Plays piano, in the UofA Folk Rock ensemble
- Looking for a job or internship for the future

# Student Introductions

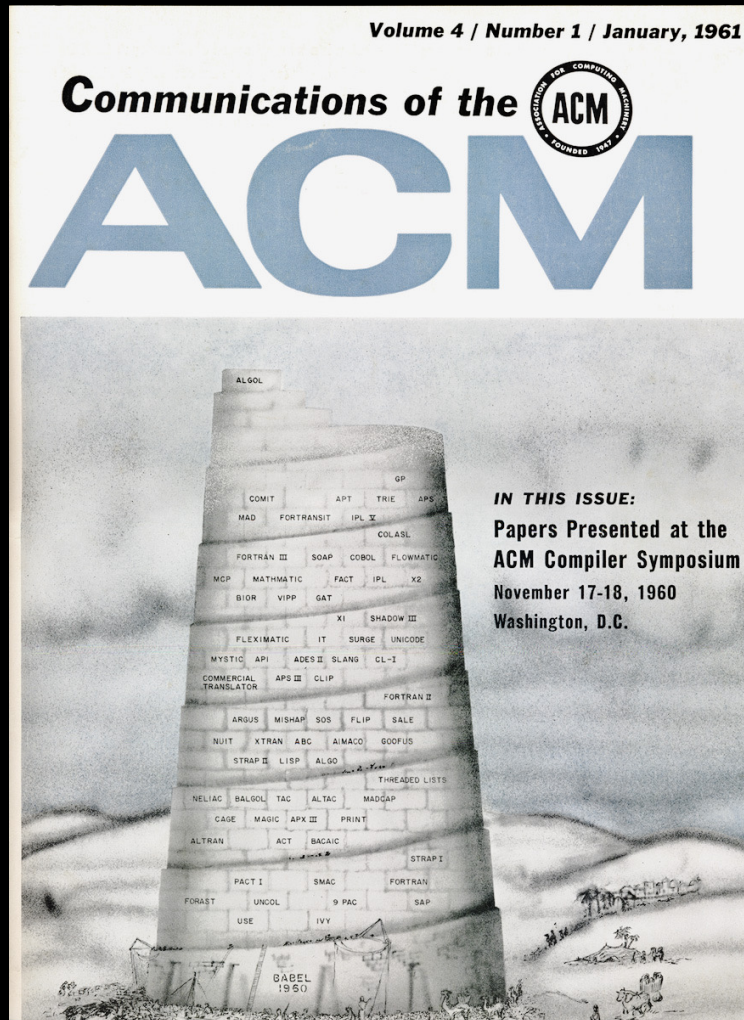
- **TopHat Questions about you and PLs**
  - How long have you lived in Tucson?
  - Programming language you feel most proficient and experienced in?
  - Languages you have programmed in for 1+ semesters?
  - Languages you have written at least one example in?
  - Languages you would like to learn more about?
  - What motivates you to want to learn a new programming language?

# Asking Questions in Class

- **Please do!!**
- **Options**
  - Raise your hand
  - Ask a question in the `in_class_questions` category in Piazza
- **In class activity**
  - Sign into Piazza for 372
  - For future classes
    - Bring a device can program on to each class
    - Also a writing utensil



# There are thousands of languages

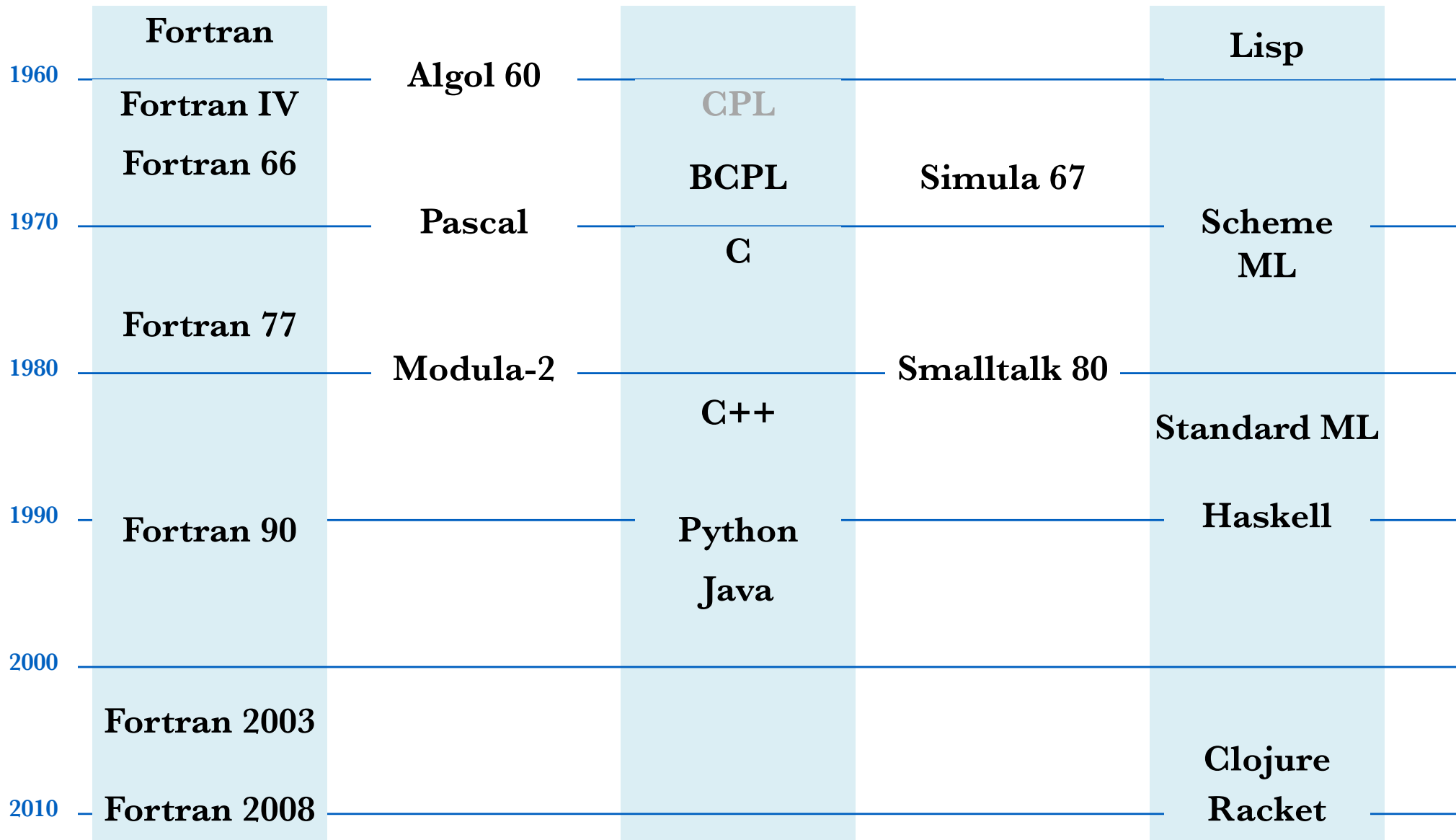


Fortran appeared around 1957,  
Lisp around 1958.

By 1959, the Tower of Babel  
analogy was being applied to  
programming languages.

# Some Language Families

## A short list of languages





- **Compare different languages and paradigms in the context of real-world problems**
  - You will need a vocabulary and an understanding of a range of programming language features to do this
  - Use these skills and concepts to build confidence in reading and understanding code in any programming language
- **Some example language features we will discuss**
  - First-class functions, i.e., lambda functions
  - Pattern matching
  - Type inference
  - Generics/templates
  - Concurrency and parallelism

# How we will study language features

- **Write small programs exercising these features**
  - Standard ML
  - Prolog
  - Chapel
- **Compare programming languages (TopHat)**
  - Evaluation criteria
  - Tradeoffs between different design choices in the context of applications

# Participation Quiz (15 minutes)

- **You will get feedback in Gradescope**
- **If you answer all the questions, you will get all of the points on the quiz**
- **It is OK if you do not know any of the answers!**
  - This first participation quiz is to collect a baseline.
  - Put your answers in the provided box.
  - If you do not have any idea, please use a question mark.

# Course Logistics (Highlights from Syllabus)

- **Out-of-class work (400/1,000 points)**

- Eight SAs (small assignments) at 15 each, capped at 100 total points
- Three LAs (large assignments) at 40 each, capped at 120 total points
- Final group project, 200 points
- SA1 on docker use is due Wednesday Jan 22<sup>nd</sup> at 11:59pm

- **In-class work (600/1000 points)**

- $\geq 120$  points work of participation and other quizzes, capped at 100
- MT1 100 points, MT2 200 points, Final exam 200 points
- Final exam score can replace MT2 score

- **To pass the course, you must earn at least 360/600 (60%) of the in-class points**

# Collaborating with Others and AI

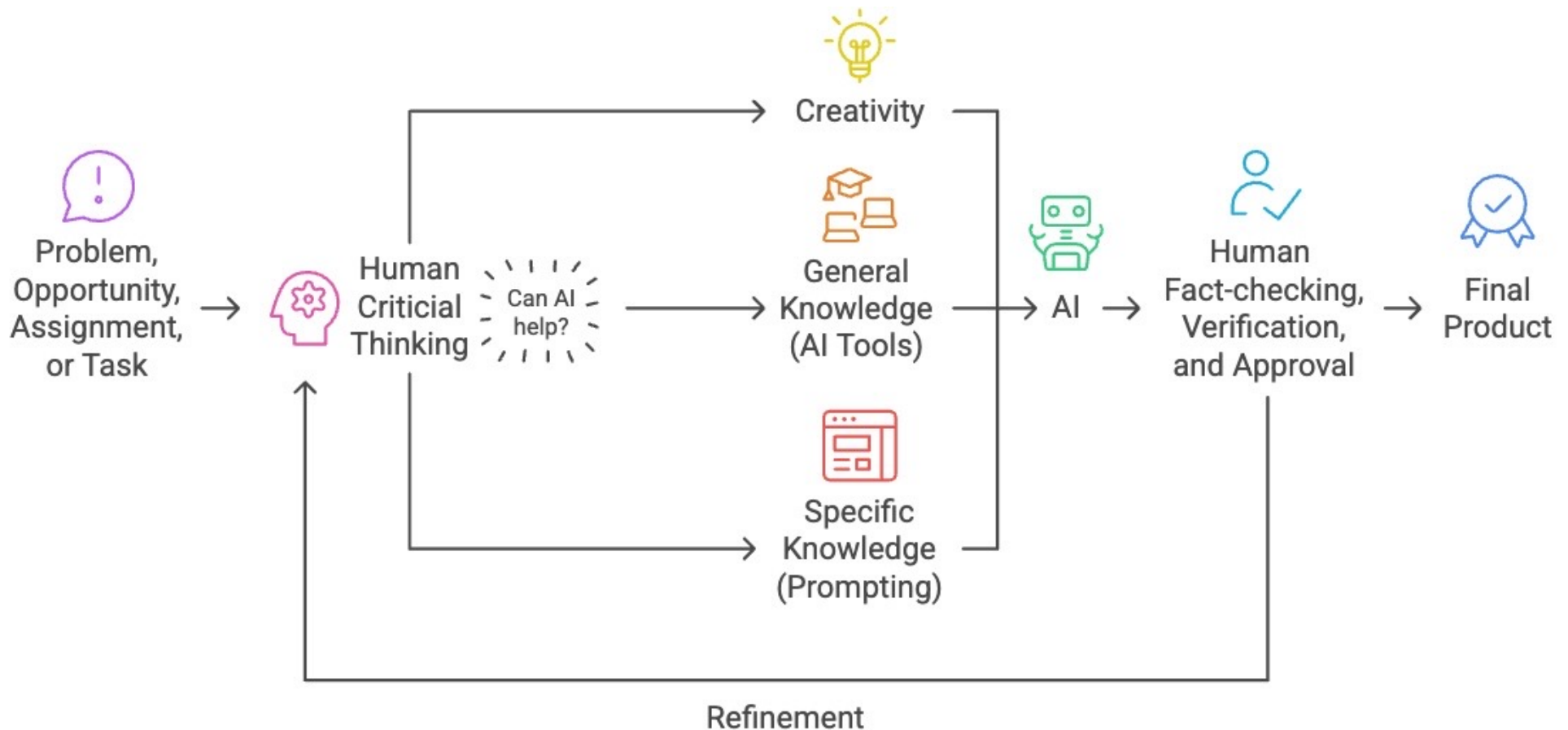
- **Highly encouraged**

- Share ideas, ask each other questions, ask free LLMs questions
- Share questions and answers about concepts
- The goal is for you to learn and be able to use the material
- My responsibility is to evaluate the extent to which you have

- **AI generators (LLMs, copilot, etc.) are tools**

- Google caused a shift in how programming was done in the 1990s
- AI generators are causing a similar shift
- Programmers will not be replaced by AIs, but they will be replaced by other programmers that know how to effectively use Ais
- Take opportunities to practice using AI generators to help you learn

# Dr. Randi Weinstein's Philosophy of AI Use



# Suggested Workflow for the Course

## • Daily

- Check the schedule in syllabus on GitHub, <https://github.com/UofA-CSc-372-Spring-2025/CSc372Spring2025-CourseMaterials/blob/main/syllabus-csc372-Spring2025.md>
- Check Piazza for announcements
- Do some spaced repetition practice with cards

## • Class days

- Bring laptop and potentially a cell phone to class
- Ask questions in class and answer TopHat polls
- Create ~5 spaced repetition cards after each class or while reading

## • Weekly minimum of 9 hours for 372 (TopHat)

- 2.5 for class, 1.5 for 20 minutes space repetition each day
- Leaves  $\geq 5$  hours for reading and current SA, LA, or FP



# Spaced Repetition

- **Anki demo (TopHat)**

- Put in some questions and answers about the syllabus
- Show what a review session looks like
- Show them questions at the bottom of Ray Toal reading

# Small Assignment 1 (SA1)

- **Posted in syllabus schedule (TopHat)**
  - Building a Docker container
  - Running programs in SML, Prolog, and Chapel in Docker
  - Editing files with an editor of your choice and seeing the changes in Docker
  - Doing GitHub commits and pushes
  - Submitting the assignment in Gradescope

# Applicable Skills for the “Real World”

- **Definition of “Real World”**

- Industry: single proprietor, startup, ... to a big company like HPE
- Labs: Argonne, LLNL, Oakridge, ...
- Academia: research, teaching, service

- **Skills**

- Collaborating with others
- GitHub, or similar tools, are used everywhere
- Actively curate your project portfolio while taking classes
- Read and understand existing code
- Compare different programming tools for use in projects where there is time, money, and credibility at stake
- Leverage AI tools

# Course Logistics that Encourage Skills

- **Extra Credit**

- Answering others in a substantive way on Piazza
- Create a study guide for quizzes, midterms, and final exam
- Do a pull request to fix an error in course materials.

- **Assignments and In-Class Activities**

- Work with others but you are evaluated individually, just like in industry and academia
- Learning new PLs and working with existing code

- **Limitations of a class format**

- Won't be able to do 1:1's with your “manager/advisor”/instructor every other week
- Figuring out what questions to ask and proposing what to do is the hard part in the real world, in a class most of that has been done

# Welcome to 372!!

- **Questions?**
- **Instructor Office Hours**
  - Every day after class in GS 707
  - Feel free to walk back to GS with me
- **TA Office Hours**
  - Phyllis, Tuesdays 1:30-2:30pm, TBD room
  - CJ, TBD

