CS534

Types and Programming Languages

Prof. Stefan Muller

Spring 2024

Programming Language =	
Syntax	What programs *look like*
+	
Semantics	What programs *mean*

Semantics =

Static Semantics

At compile time

Can we assign meaning to a program without running it?
Type systems
Static analysis
Verification (see CS536)

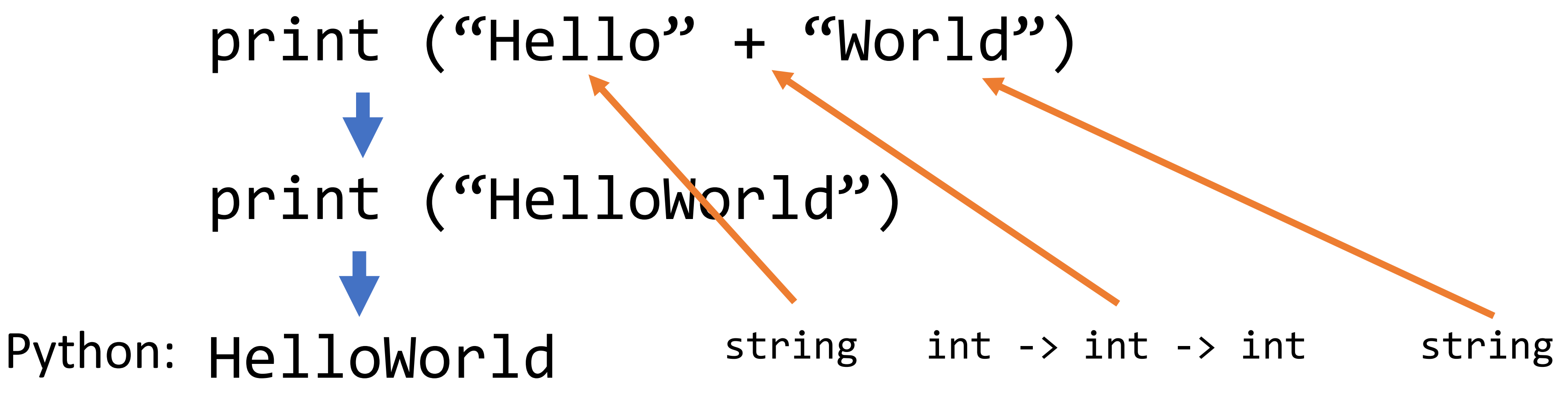
+

Dynamic Semantics

At run time

How do we evaluate the program?

What are the semantics of this program?



```
print (“Hello” + “World”)

print (“HelloWorld”)
```

Python: `HelloWorld`

OCaml:

```
Error: This expression has type string but an
expression was expected of type int
Error: Unbound value print
```

What are the semantics of this program?

```
print ((“Hello” + “World”) + 42)
   ↓
print (“HelloWorld” + 42)
   ↓
```

Python: `TypeError: can only concatenate str (not "int") to str`

Type systems can do more than that

Does this program look right?

```
str = input_string();                 string[private]
if (string_len(str) > 10) {
  print(str);                         string[public] -> void
} else {
  print(“Too short\n”);
}
```

Type systems can do more than that

```
function compute (filename) {
  file = open(filename);
  …
}
```

file isn’t in the cloud!

```
(string -> unit)[server] -> unit      (string -> unit)[local]
```

```
function main () {
  …
  save_to_file(“data.dat”, data);
  cloud_api.compute_on_cloud(compute, “data.dat”);
}
```

Developing type systems is hard

- Fixing the password leak:
  - Functions that print can only take a public string
  - Fixed, right?

```
string[private] password = input_string();
string[public] str = password;
print(str);
```

Developing type systems is hard

- OK, let’s try this again. You can’t write a private string to a public string either.

```
string[private] password = input_string();
string[public] str = new string();
for (int[public] i = 0; i < MAX_PWORD_LEN; i++) {
  for (char[public] c = ‘a’; c <= ‘z’; c++) {
    if (c = password[i]) {
      str[i] = c;
      break;
    }
  }
}
```

Or, if this is C:

```
str = (string[private])password
```

How do we know we didn’t make a mistake?

- Proofs

Theorem:
If a program P takes a private input and returns a public output
and P is well-typed
and s and s’ are different private inputs,
then P(s) = P(s’)

Proof:
(Maybe later in the semester)

Type systems/PL techniques aren’t just for programming languages

(yes, I have a PL paper for each of these)

- Distributed/Heterogenous systems
- Garbage collection
- Information security
- Memory usage/execution time
- Priority Inversions in Interactive Programs
- Modeling Unix Pipelines
- Networking
- Law
- Knitting
- Computer-aided Design (CAD)
- Music
- Roller Coasters
- Generative Interactive Systems (think text adventure games)

Lots of potential for advanced lectures/guest lectures throughout the semester

This course (rough schedule)

- Important techniques in PL
- Modeling imperative programming languages
- Modeling functional programming languages
- **Midterm (early March)**
- Polymorphism, Parametricity and Abstraction
- Recursion and Recursive Types
- Subtyping
- Models of Evaluation
- Advanced topics (based on time and interest)

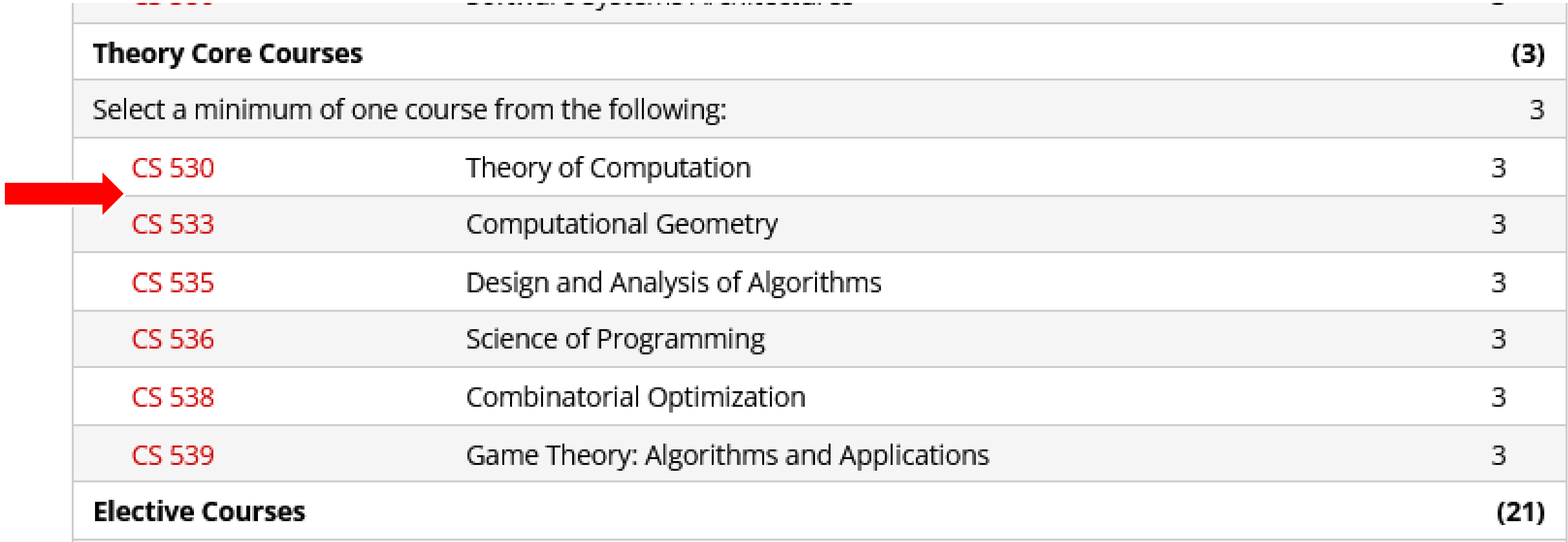
Logistics

- Course website: http://cs.iit.edu/~smuller/cs534-s24/
  - Includes tentative schedule
  - Lecture notes will be posted there
  - Check it frequently!

- Blackboard for assignments
  - Also lecture videos

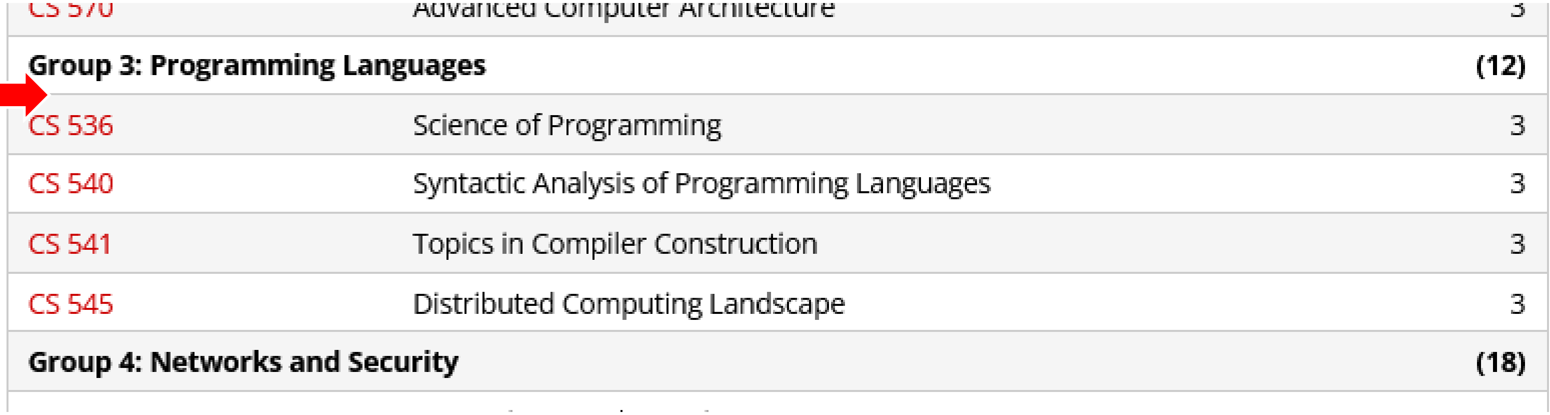
- Discussion platform: Piazza
  - https://piazza.com/class/lr437g5ugf93n7/#

More logistics



Theory Core Courses		**(3)**
Select a minimum of one course from the following:		3
CS 530	Theory of Computation	3
CS 533	Computational Geometry	3
CS 535	Design and Analysis of Algorithms	3
CS 536	Science of Programming	3
CS 538	Combinatorial Optimization	3
CS 539	Game Theory: Algorithms and Applications	3
Elective Courses		**(21)**

- MCS/MSCS – Theory requirement
- PhD – Languages (Group 3)
  - Written Qualifer?
    - Some confusion.
    - I'll let you know



CS 570	Advanced Computer Architecture	3
Group 3: Programming Languages		**(12)**
CS 536	Science of Programming	3
CS 540	Syntactic Analysis of Programming Languages	3
CS 541	Topics in Compiler Construction	3
CS 545	Distributed Computing Landscape	3
Group 4: Networks and Security		**(18)**

More logistics

- It’s been approved by the CS Graduate Studies Committee, but hasn’t made it into all the right systems (DegreeWorks, bulletin) yet.
  - Your advisor may need to add a petition on DegreeWorks to count this class for the requirement. Tell them to contact me with any concerns.

Prerequisites

- Officially: CS430 with a min. grade of C
- Informally:
  - Familiarity with basic logic
  - Comfort with mathematics, formal notations
  - *Some* programming experience in a statically typed language
  - *Some* experience programming with recursion

  - The first homework will have some questions that test the above background knowledge. If you’re not comfortable, set aside time to review it on your own or reconsider taking the class.

More about that programming requirement

- There will be no (real) programming in the class.

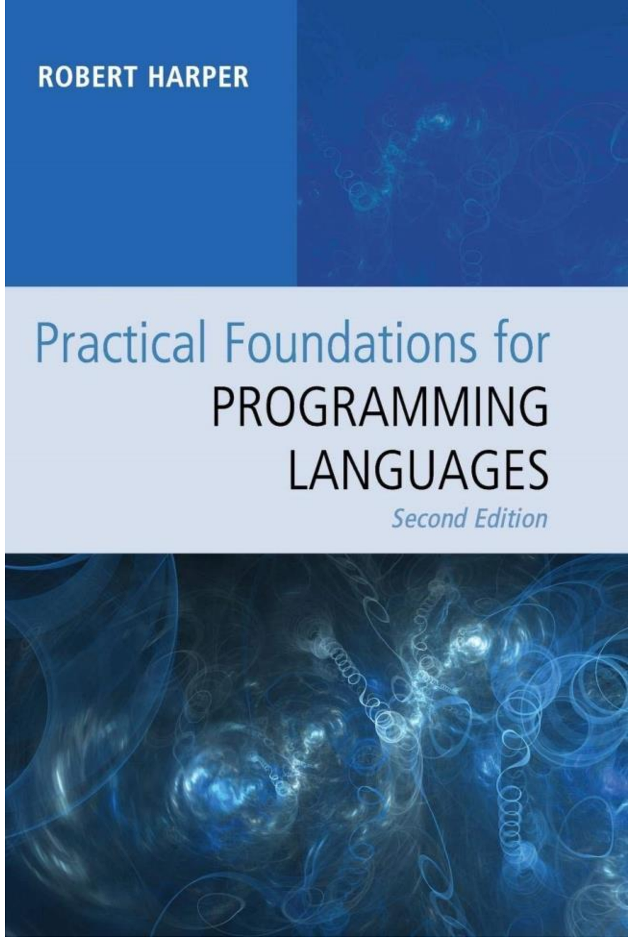
- Mostly theory: there **will be** proofs
  - Proofs *about* programs: we will be using some small languages
    - IMP: a small language with assignment, if/then/else, while, etc… (see CS536)
    - Some small functional programming languages (like Haskell, OCaml, etc.)
  - May have to write some small programs in these small languages
  - Need to be familiar with the mechanics of programming and types

Recommended Textbook

- *Practical Foundations for Programming Languages, Second Edition*
- by Robert Harper (Cambridge, 2016)

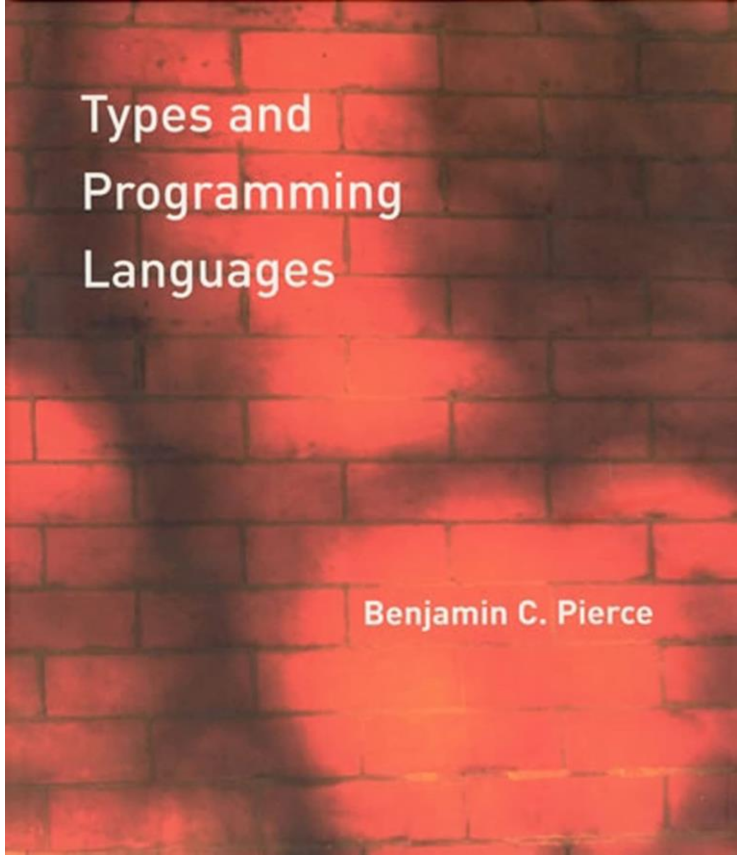
- Available from the campus textbook store
- Or online as a preprint (link on the website)

- Not necessary to buy it: lectures intended to be self-contained, but this is a good reference if you want one



Alternate Textbook

- *Types and Programming Languages*
- by Benjamin Pierce (MIT, 2002)



Grading

- 35% Assignments
  - Roughly 7, may not be evenly weighted
  - HW1 is up on Blackboard, due Thursday, 1/18
- 25% Midterm exam
- 35% Final exam
- 5% Participation/discussion
  - Do some (or some combination) of:
    - Answering/asking questions in class
    - Answering/asking questions on Piazza
    - Attending office hours

Exams

- Midterm: 75 minutes, normal class time
- Final exam during finals period (date set by Registrar)
    - Final exam *is* cumulative
- Some kind of notes allowed
- Sections 01, 02: In-person
    - If you CANNOT take the exam in person, let me know
- Section 03: Will get an email discussing options
    - Preferred: take with the in-person students.
    - Also possible: take somewhere else with a proctor

Late Days

- 8 late days per student
- Each late day extends the deadline 24 hours
- Can use <= 2 per assignment
  - Can't use on exams
- No need to tell me you're using them, just submit <= 2 days late
- After late days used up: 10% penalty per day late
- **No work accepted >2 days late** without prior instructor approval

Submitting Homeworks

- Blackboard -> Assignments -> HW*n*
  - Download and submit
  - The “Due Date” on Blackboard is without late days.
  - If you use late days, it will be marked as “LATE.” Don’t worry about this.

- Submit a .doc, .docx, or .pdf
- Heavily preferred (and 3% bonus points): Use LaTeX to generate a pdf
  - Template on Blackboard
  - Resources on course website

Academic Honesty

- All submitted work (homework and exams) is to be your own individual work unless specified otherwise.
- Specifically prohibited (but this list isn’t exhaustive):
  - Sharing answers with other students (even if one or both of you is done)
  - Looking online for answers
- Specifically permitted:
  - Getting help from the instructor
  - Getting help from the ARC or other official university tutoring resources
    - If you want to use an outside tutor, let me know first

Academic Honesty

- Penalties (for every violation):
    - Zero on the homework or exam
    - Report to academic honesty
        - May result in university-level sanctions after first report

Generative AI (e.g., ChatGPT)

- For most parts of most assignments:
  - OK to use it, but you **must** indicate **specifically where and how** you used it
    - Especially: indicate any text that came directly from ChatGPT
    - If you took text and edited it, explain how
  - If possible (**in addition to the above)**, attach chat logs
  - Examples:
    - "On Task 3.2, I asked ChatGPT to explain the dynamic semantics of the untyped lambda calculus for me, but I did the rest myself"
    - "The bolded text came directly from ChatGPT. The italicized text came from ChatGPT but I edited it. ChatGPT's original text appears below."
    - "I wasn't sure about my answer to task 2.2, so I asked ChatGPT some questions. I marked in bold the parts of my answers I changed based on what it said."

Generative AI (e.g., ChatGPT)

- For most parts of most assignments:
  - OK to use it… (see caveats on previous slide)
  - This isn't a trap! If an assignment doesn't say not to use ChatGPT, I won't get you in trouble for any properly disclosed use of it.
- However:
  - I'll grade your assignment exactly the same (you're responsible for checking/editing anything it gets wrong! It gets **a lot** wrong!)
  - For some assignments/questions, I'll clearly indicate that it's not allowed
  - Undisclosed use of Generative AI or use on questions where it's not allowed is a violation of the academic honesty policy
  - This policy is subject to change

Does that mean I can just have ChatGPT do all my homework?

- HW1:
  - 49 points you're allowed to use ChatGPT on (not allowed on the questions about background knowledge)
  - ChatGPT's score: ~22/49

	Max	**ChatGPT**
Homework	35	17
Midterm	25	5
Final	35	7
Participation	5	5
Total	**100**	**34**

Course Staff

- **Instructor:** Stefan Muller

- Office hours (info including links will be posted):
    - In general (after next week): Mon., 10-11 on Zoom; Thur., 2-3pm SB 218E
    - This week: Fri., 1-2 on Zoom
    - Next week: Mon., 10-11 on Zoom, another TBA

- I am here to answer your questions! Really!

Other ways to get help

- Piazza
  - I’ll email a link

- Academic Resource Center (ARC): www.iit.edu/arc
  - FREE subject matter tutoring and academic coaching

	Piazza	**Office Hours**	**Email**	**ARC**
General questions about lectures, logistics, etc.	✓	✓		
General discussion, clarifications, about HW questions	✓	✓		
Specific questions about your HW answers		✓		✓
More in-depth personal tutoring				✓
Personal matters (accommodations, other requests, etc.)			✓	

Using Piazza

- Post questions about lecture, homeworks, logistics, etc.
  - Just don’t give away your answers to HW questions
  - Posts grouped by topic
- Anyone can answer-I’ll check student answers and fix them if needed
- You can be anonymous (just not to me)

- Private posts (just to me) – enabled for now but not much use
  - Asking about grades, personal matters? Email instead
  - Asking specific questions about your homework answers? Come to office hours