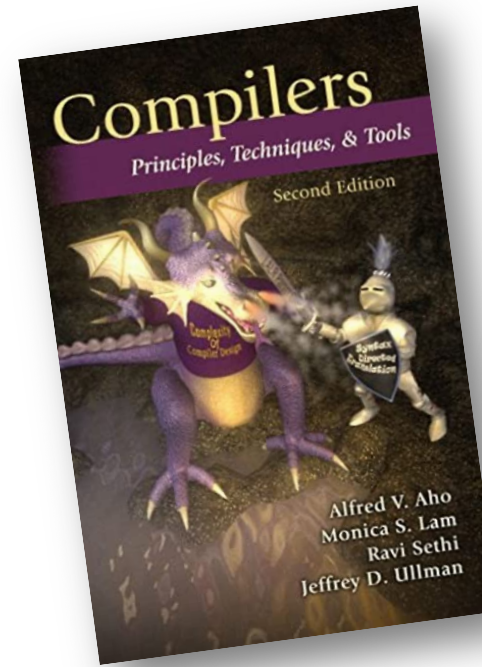


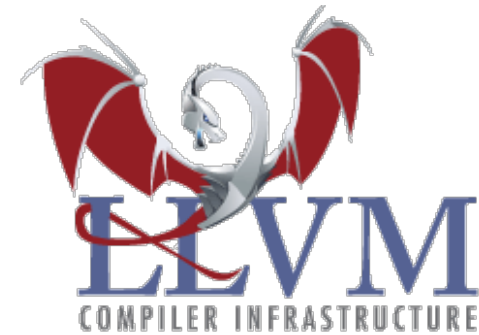
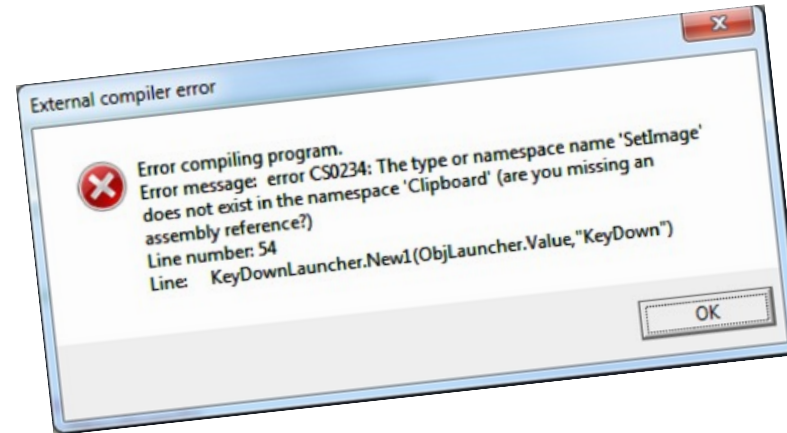
CSE211: Compiler Design

Sept. 22, 2022

- **Topic:** Course Introduction
- **Questions:**
 - *What is a compiler?*
 - *What are some of your favorite compilers?*
 - *Have you ever built a compiler?*



```
<expr> ::= <term> "+" <expr>
          | <term>
<term>  ::= <factor> "*" <term>
          | <factor>
<factor> ::= "(" <expr> ")"
          | <const>
<const> ::= integer
```



Enrollment

- Huge issue with CSE 110A prereq
 - Hardly anyone can register
 - I've had to manage an informal waitlist through my email
- The class has a capacity cap of 19
 - Can't over enroll due to health/safety reasons
- I have messages from 26 of you. It means approximately 37% of people who want to be in the class will not be able to get in.
 - I wish I could accept everyone
 - Roughly the cutoff seems to be the first week of August

Enrollment

- Please email me after class. I will cross reference your email with your initial request and see where it falls on the waitlist.
- If I don't hear from you by midnight tonight, I will assume that you no longer are interested in a spot in the class.
- I am working with the department so that this class will have an official waitlist in the future.



<https://users.soe.ucsc.edu/~tsorensen/>

Room E2-233

- Tyler Sorensen
third year faculty at UCSC
third time teaching this class
First time fully in person
- From rural Utah
- Ph.D. at Imperial College London
- Work on parallel programming languages
- Invited member of the Khronos Group

Are things back to normal?

Special Note for Fall 2022: COVID continues to effect our lives in many different ways. There will likely be infections throughout the year (both as students and faculty). Additionally, we will all have to adjust as we work together to create an environment where everyone feels valued and safe. The pandemic has been difficult for people in different ways and there will surely be new challenges throughout the quarter. We will work together to overcome them and make this a productive and engaging class. That said, this class is designed to be *in person*. If you do not plan to make a majority of the lectures in person, then I would recommend reaching out to a graduate adviser to find a class more suitable to your constraints. Please see the COVID policy in the [overview](#) and email me if you have any questions or comments.

Today's class

- Class syllabus (I apologize in advance for the text slides)
- Jumping into module 1
 - Tokenizing

Course resources

- Public course website:
<https://sorensenucsc.github.io/CSE211-fa2022/index.html>
 - Schedule, slides, syllabus, additional resources
- Private course website: Canvas
 - grades, announcements, SETs, homeworks, tests, lecture recordings
 - intro quiz
- Docker Image
 - Used for homework (instructions incoming)

Description

In this class you will learn about advanced topics in compiler design and implementation. In the abstract, compilers explore many of the [foundational problems in computer science](#). In practice, compilers are [massive pieces of well-oiled software](#), and are some of the engineering marvels of the modern world. Given the end of Dennard's scaling, compilers will play an increasingly important role to achieve further computational gains. *The main focus of this class is how compilers can make your code more efficient and safe on modern (and near-future) processors.*

Background

- Official prereq is an undergrad UCSC class (CSE 110A)
- But this is a graduate course...

Background

- Understand basics of parsing:
 - Regular expressions
 - Context free grammars
- Understand basics of programming languages
 - how are programs structured (e.g. basic blocks, expressions, statements)
- Comfortable using console coding (e.g. emacs or vim) and a terminal
 - Python
 - C/C++
- Some exposure to architecture, especially machine code (e.g. MIPS or X86 or RISCV)

Background

- First module and assignment will cover lots of compiler basics

Modules

- There are 5 modules
- We will aim to spend roughly two weeks on each
 - We will be flexible. This is a small class and if we need to spend more time on a topic, we will!
- The last module may be cut short
- Structure
 - first 3 modules will be given lecture style
 - last 2 modules are more like a reading group/discussion

Modules

Module 1: Parsing Overview: This module will go over parsing at a high-level. This includes tokenizing, parsing context-free grammars, parser generators, and how to quickly create a parser for a simple programming language.

This is not a class on implementing parsers! Instead I want you to learn how to easily implement a compilers using parser generators!

Modules

Module 2: Analysis and Optimization: This module will go over different flow analysis (aka static analysis). We will discuss different AST traversals, SSA form, and applications such as identifying use-before-initialized errors.

Modules

Module 3: Parallelization and DSLs: We will discuss how compilers can be used to transform sequential code blocks into equivalent forms that can be executed in parallel. We will explore domain specific languages that further facilitate this automatic translation.

Modules

Module 4: Advanced Topics: We will read impactful papers in the area and discuss them. We will adjust based on class interest, but topics may include optimization evaluation and synthesis.

Modules

Module 5: Final Project Presentation and Guest Lecture: The last few classes will have presentations from final projects and a guest lecture (if there is time).

Class Format

- in-person
- 95 minutes
- *Non-protected materials* are public
- *Protected materials* are on canvas
- *A few classes will be asynchronous due to travel, see the schedule.*

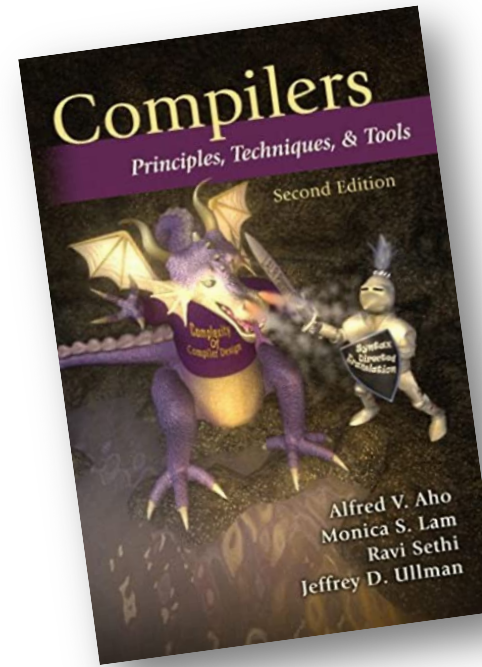
Class Format

- In-class discussion questions on start of each lecture slide
- If the class room is available, I will try to stay 15 minutes after.

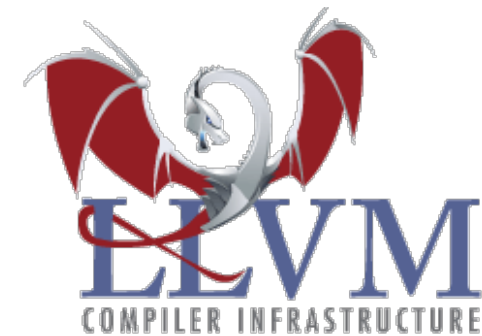
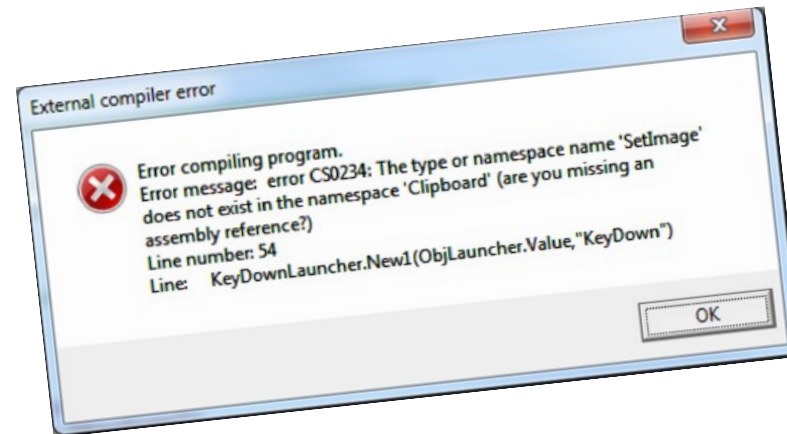
CSE211: Compiler Design

Oct. 1, 2020

- **Topic:** Course Introduction
- **Questions:**
 - *What is a compiler?*
 - *What are some of your favorite compilers?*
 - *Have you ever built a compiler?*



```
<expr> ::= <term> "+" <expr>
          | <term>
<term>  ::= <factor> "*" <term>
          | <factor>
<factor> ::= "(" <expr> ")"
          | <const>
<const> ::= integer
```



Class Format

- We will use Piazza for discussions
 - I will check and try to reply within 24 hours to discussion questions
 - Not guaranteed to reply in off-hours or weekends
 - Please start class-wide discussions, link to interesting articles, etc. etc.
 - Please DO NOT publicly share homework solutions before the due date or ask for detailed help in the class-wide channels. Ask privately.
 - I will moderate.

Attendance

- Small class means lots of discussion! Please try your best to attend
- Message me before hand if you will miss a lecture
- 17 out of 20 lecture attendances required; more than that will be evaluated on a case-by-case basis.
- Please participate! In-person, Piazza discussions, interacting with your classmates, etc.

Office Hours

- Wednesday from 3 – 5 pm Pacific Time.
 - Remote or in-person
 - I will create a google doc sign up sheet with a zoom link
 - Room is E2 233
- Post clarifying questions as discussion topics so that everyone can benefit!

Homeworks

One assignment for the first 4 modules

- 2 weeks for each homework, posted midway through module.
- Each assignment has a suggested language. If you want to use a different language, please approve with me
 - Final solution must run in the docker and have exactly the same the interface
 - Using a different language may require more documentation
 - But it can be a fun exercise (especially if you use a functional language!)
- Ask questions on Piazza, discuss concepts, visit office hours, etc.
- Start early!

Tests

- Designed to take ~120 minutes (not including studying)
 - No strict timer
 - Midterm will have 1 week
 - Final will have 1 day (8 am to 8 pm)
 - Scheduled time for final is: 8 am to 11 am
 - ***Often students take longer! Plan accordingly***
- Open book and open note
- Please:
 - Do not collaborate with classmates
 - Do not google for exact answers (googling for concepts is fine)
 - Ask any clarifying questions through private piazza posts to me

Paper Assignment

- Propose an academic paper of interest (or I can suggest one) no later than 3 weeks before the end of classes.
- Write a 4 page double spaced review of the paper (due before the final)
- List of papers:
 - https://docs.google.com/spreadsheets/d/1C5zMXC_RC94EmgmJVOLIwYLylEyDUmK9KCO8oixaC3k

Final Project

You can choose to do a final project instead of a final exam

- Final projects must be approved by me no later than 2 weeks before the end of the semester
- 6 page double spaced report on the project (due at the date of final)
- 15 minute presentation to the class (due day of class)
- If you are a grad student, I suggest thinking about a project you can incorporate into your thesis/project.

Final Project

Option to publish final project online:

<https://sorensenucsc.github.io/CSE211-fa2021/projects.html>

Rubric

- 10% Attendance
- 40% Homework (8% each)
- 10% paper assignment
- 10% Midterm
- 30% Final

Letter Grade	Percentage	GPA
A+	97–100 %	4.33 or 4.00
A	93–96 %	4.00
A–	90–92 %	3.67
B+	87–89 %	3.33
B	83–86 %	3.00
B–	80–82 %	2.67
C+	77–79 %	2.33
C	73–76 %	2.00
C–	70–72 %	1.67
D+	67–69 %	1.33
D	63–66 %	1.00
D–	60–62 %	0.67
F	0–59 %	0.00

From: https://en.wikipedia.org/wiki/Academic_grading_in_the_United_States

Resources

- Slides and lectures:
 - I will post the slides for each lecture by the day after
- Textbooks:
 - **Engineering: A Compiler 2nd Edition**: unlimited online availability from the library. Links in Resources tab of webpage.
 - **Compilers: Principles, Techniques, and Tools 2nd Edition (Dragon Book)**: Limited availability from the library. Ask me for links, or to borrow physical/kindle editions
- Academic papers/blogs:
 - I will link to them as needed on the Resources/schedule site

Walls of text incoming...

Academic Integrity

- One of the joys of university life is socializing and working with your classmates. I want you to make friends with each other and discuss the material. This is an advanced topics course; there is a high chance that your classmate will be your colleague throughout your career!
- **That said, I expect all assignments (homeworks, tests, paper reviews, presentations, final projects) to be your own original work.**
- If you work together with a classmate on an assignment, please mention this, e.g. in the comments of your code. If you use a figure you didn't create in a presentation, then it needs a citation. Please review the [universities policy on plagiarism](#)
- This class has a zero tolerance policy on cheating. Please don't do it. I would much rather get a hundred emails asking for help than have to refer anyone for academic misconduct.

Privacy

We will record lectures

- Things you do or say will be recorded. I doubt that this will be an issue, but if you want me to remove any part of the recording, please just let me know.
- Canvas is secure[™]: other communications are less so
- Let's do our best to protect our privacy and respect the privacy of others.

Disability Accommodation

UC Santa Cruz is committed to creating an academic environment that supports its diverse student body. If you are a student with a disability who requires accommodations to achieve equal access in this course, please submit your Accommodation Authorization Letter from the Disability Resource Center (DRC) to me by email, preferably within the first two weeks of the quarter. I would also like us to discuss ways we can ensure your full participation in the course. I encourage all students who may benefit from learning more about DRC services to contact DRC by phone at 831-459-2089 or by email at drc@ucsc.edu.

COVID Policy

- Please be courteous of others
 - I will be wearing a mask at least for a little bit
- Notify me directly if you need to quarantine (or take time off for other covid-related issues) so we can plan
- We will work together and have a great quarter!

COVID Policy

- COVID continues to effect us in many ways. This quarter will surely have challenges related to this. I imagine people will continue to get sick, whether it is us, or people that we love. I hope we can all approach our interactions with empathy and understanding. I will do my best to accommodate the various individual challenges that may arise. Please communicate with me early and often!
- Currently, this is designated as an *in-person* class. It is **not** a hybrid class. As explained in the [attendance](#) section, I do expect you to plan on primarily attending in person. If you cannot attend in person for reasons related to COVID (e.g. if you or someone close to you gets sick), let me ASAP to discuss accommodations. You will not lose attendance points in these cases and I will do my best to provide lecture materials to you.
- If I am unable to attend (e.g. if I get COVID), then I will aim to teach remotely as long as I am feeling well enough.

Now back to something a little lighter...

LSD Seminar

- CSE-2800-01: LSD Seminar (co-organized with Lindsey Kuper)
 - Friday's at Noon
 - 45 minute talk + 15 minutes of social
 - Student speakers!
 - <https://lsd-ucsc.github.io/lsd-seminar/2022fa/>
- Topics will be highly relevant to this class; please consider joining!

Website tour

- Class website
- Canvas

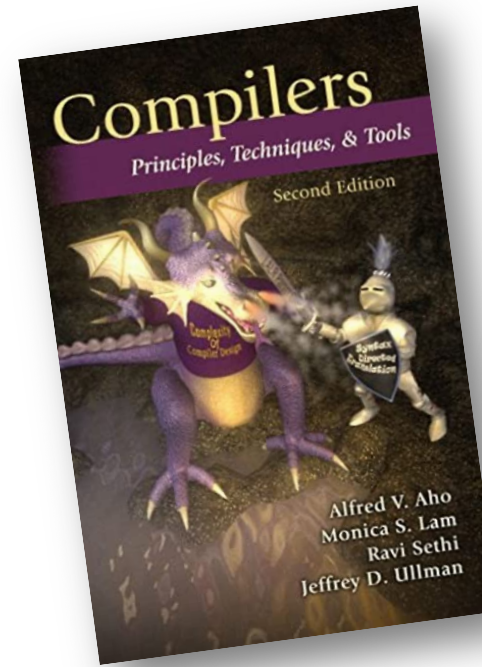
Made it through the syllabus...

The slides should start to get more interesting now

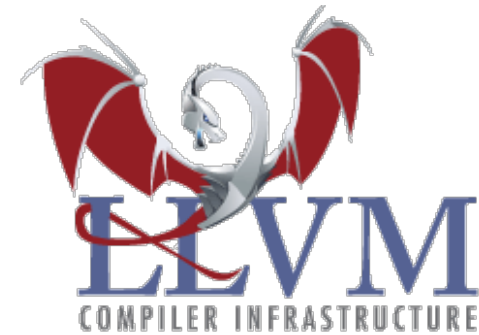
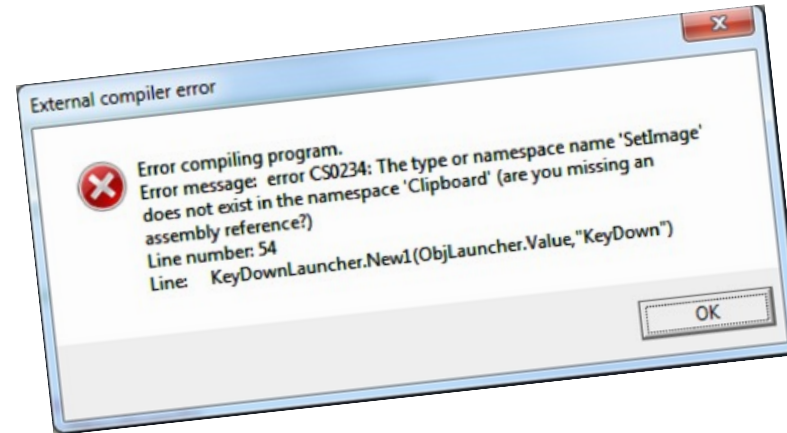
CSE211: Compiler Design

Sept. 22, 2022

- **Topic:** Course Introduction
- **Questions:**
 - *What is a compiler?*
 - *What are some of your favorite compilers?*
 - *Have you ever built a compiler?*



```
<expr> ::= <term> "+" <expr>
          | <term>
<term>  ::= <factor> "*" <term>
          | <factor>
<factor> ::= "(" <expr> ")"
          | <const>
<const> ::= integer
```



What is a compiler?

What are some of your favorite compilers

```

1 ---
2 title: "Graduate Compiler Design"
3 layout: single
4 ---
5
6
7 ### Welcome to CSE211: Graduate Compiler Design, Fall 2021 Quarter at UCSC!
8
9 - Instructor: Tyler Sorensen
10 - Time: MWF 4:00 - 5:05 pm
11 - Location: Thimann Lab 101 (in person!)
12 - Contact: <first name>.<last name>@ucsc.edu
13
14
15 Hello! I'm Tyler and welcome to the graduate compiler design course!
16
17 In this class you will learn about advanced topics in compiler design and implementation. In the abstract, compilers explore many of the foundational problems in computer science. In practice, compilers are massive pieces of well-oiled software, and are some of the engineering marvels of the modern world. Given the end of Dennard's scaling, compilers will play an increasingly important role to achieve further computational gains. The main focus of this class is how compilers can make your code more efficient and safe on modern (and near-future) processors.

```

CSE211, Fall 2021 Home Overview Schedule Homeworks References

Graduate Compiler Design

Welcome to CSE211: *Graduate Compiler Design*, Fall 2021 Quarter at UCSC!

- **Instructor:** [Tyler Sorensen](#)
- **Time:** MWF 4:00 - 5:05 pm
- **Location:** Thimann Lab 101 (*in person!*)
- **Contact:** <first name>.<last name>@ucsc.edu

Hello! I'm Tyler and welcome to the graduate compiler design course!

In this class you will learn about advanced topics in compiler design and implementation. In the abstract, compilers explore many of the [foundational problems in computer science](#). In practice, compilers are [massive pieces of well-oiled software](#), and

- Building this website started with:
- Markdown to describe the page
 - compiled with Jekyll to a static webpage
 - static webpage is in HTML and javascript

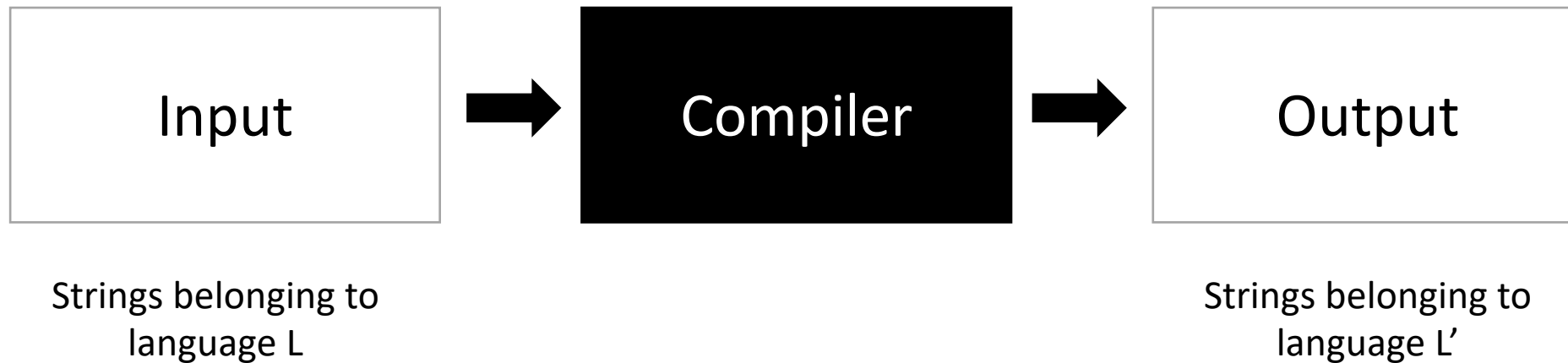


Have you ever built a compiler?

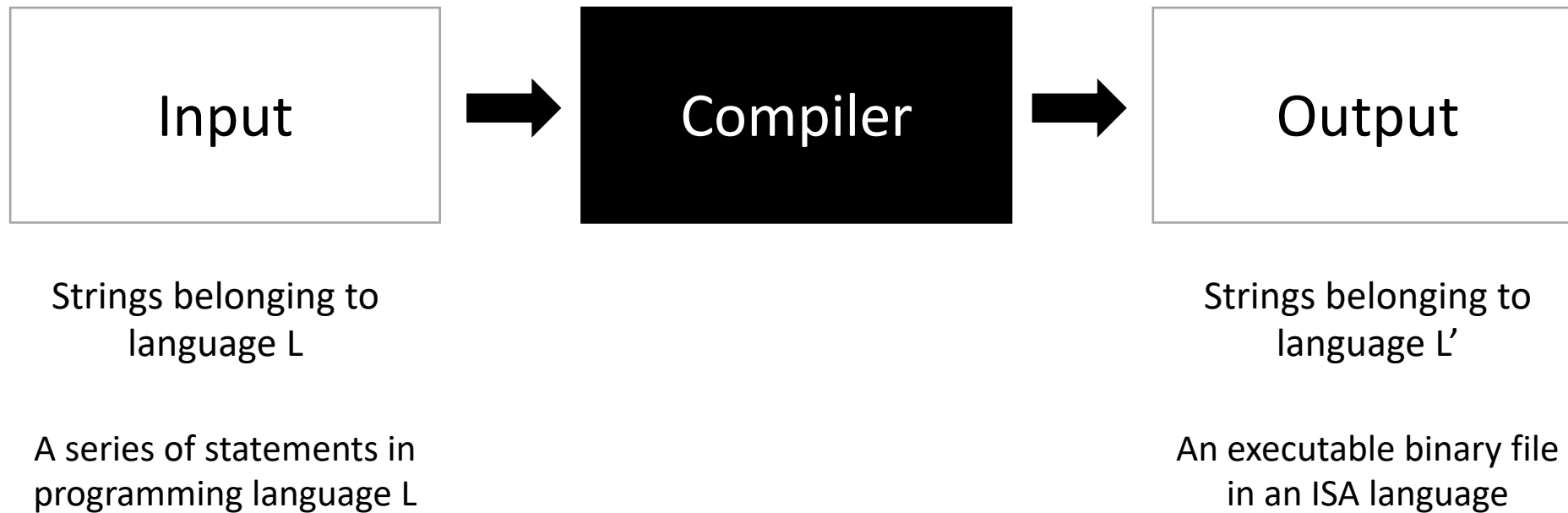
What is a compiler?



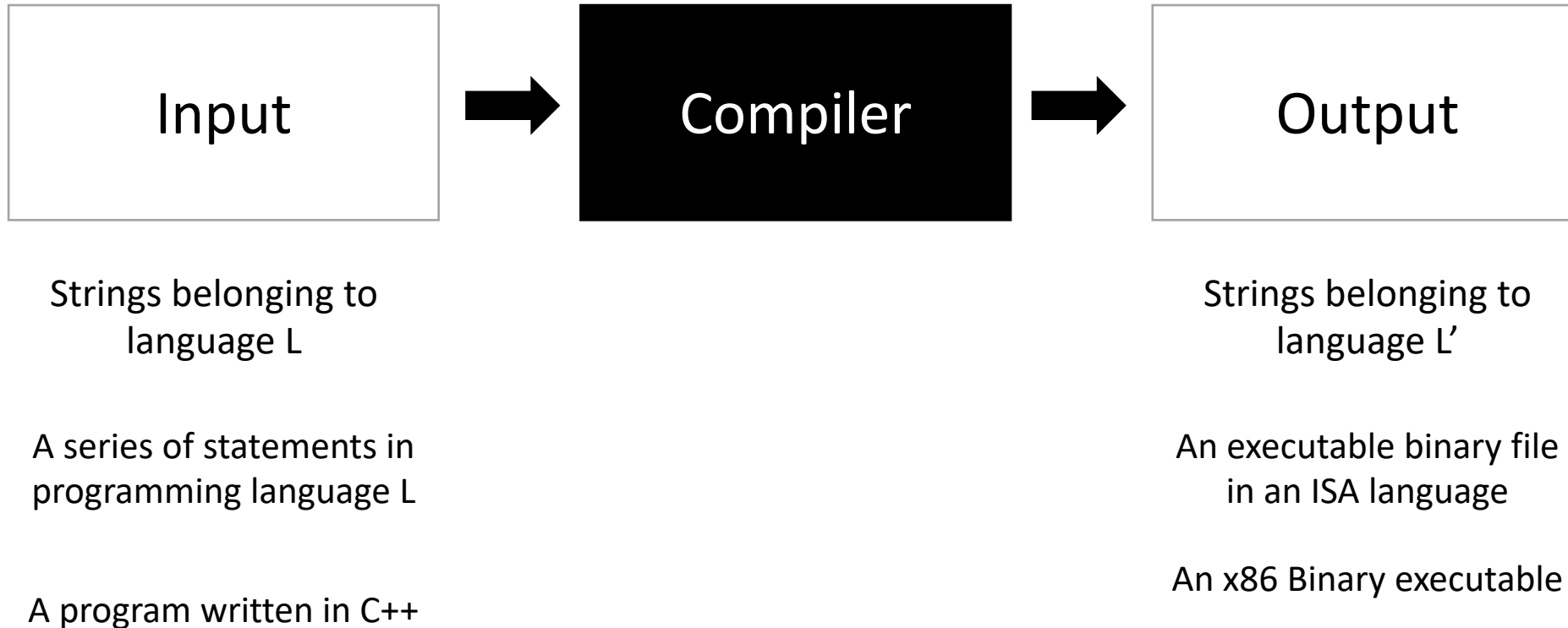
What is a compiler?



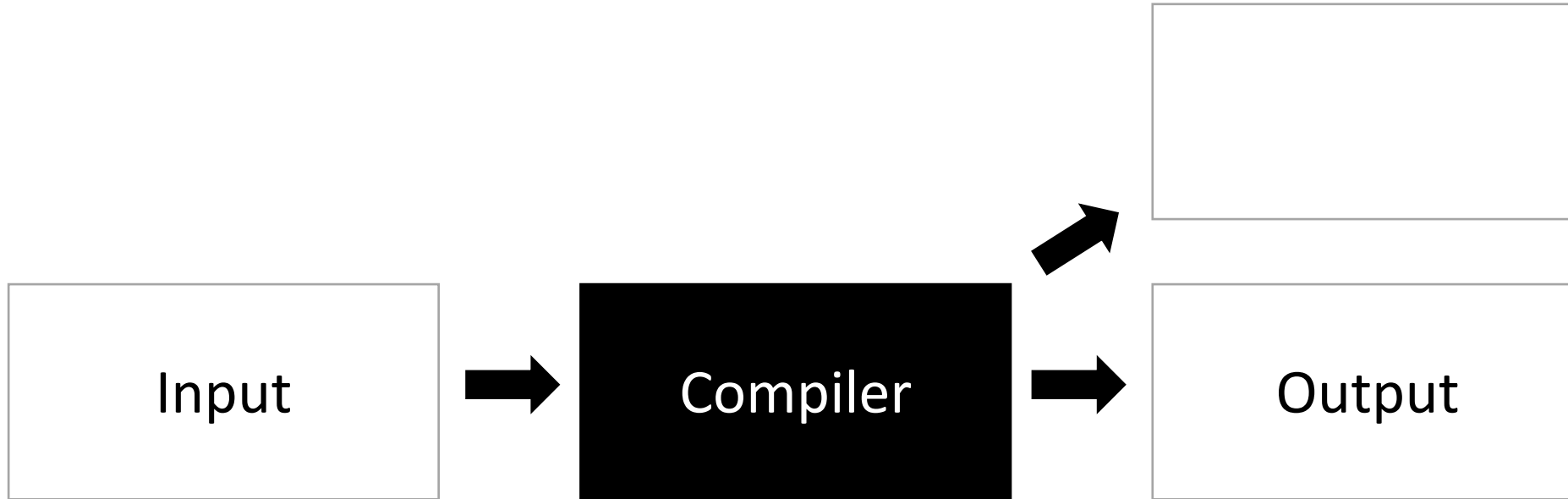
What is a compiler?



What is a compiler?



What is a compiler?



Strings belonging to language L

A series of statements in programming language L

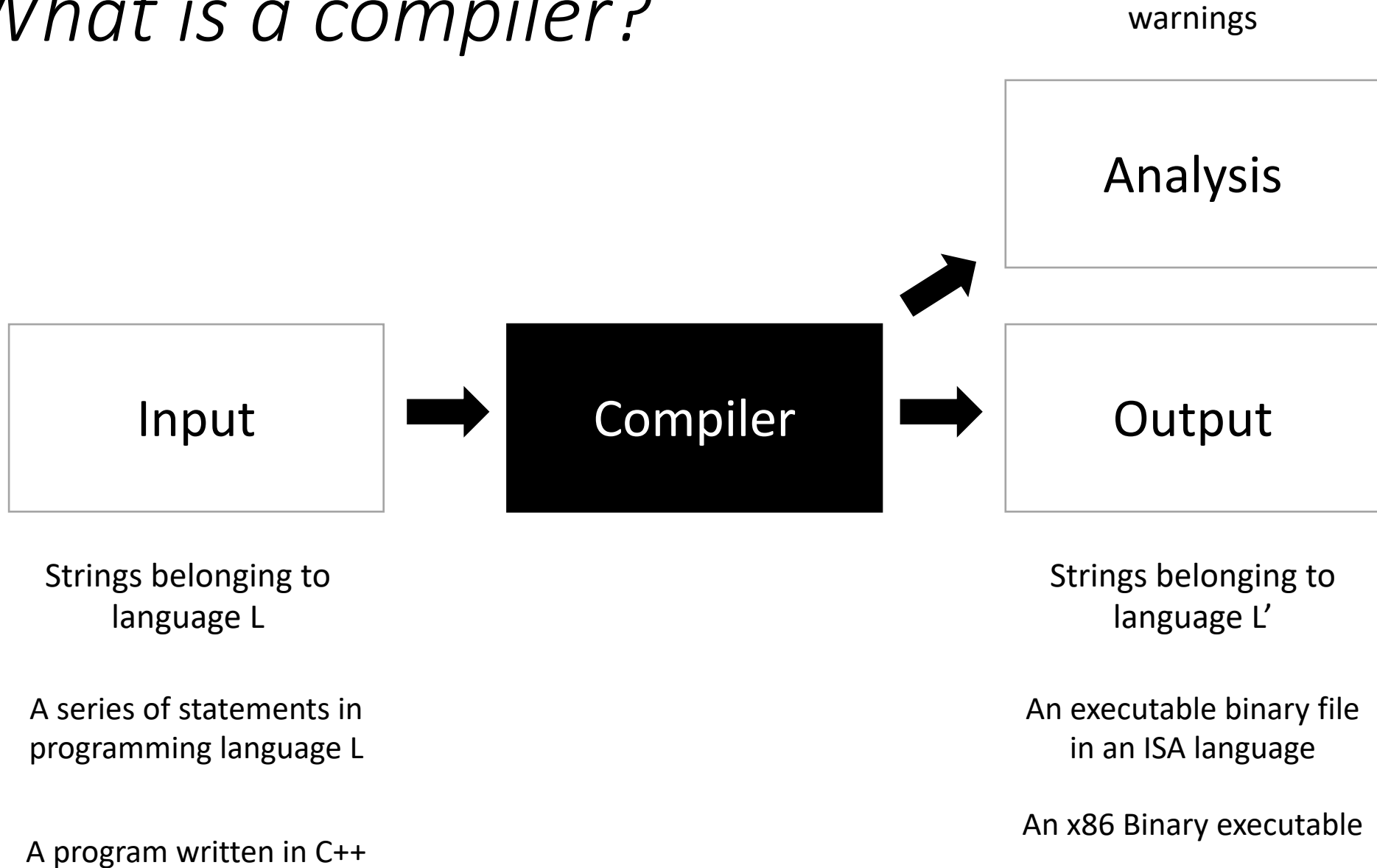
A program written in C++

Strings belonging to language L'

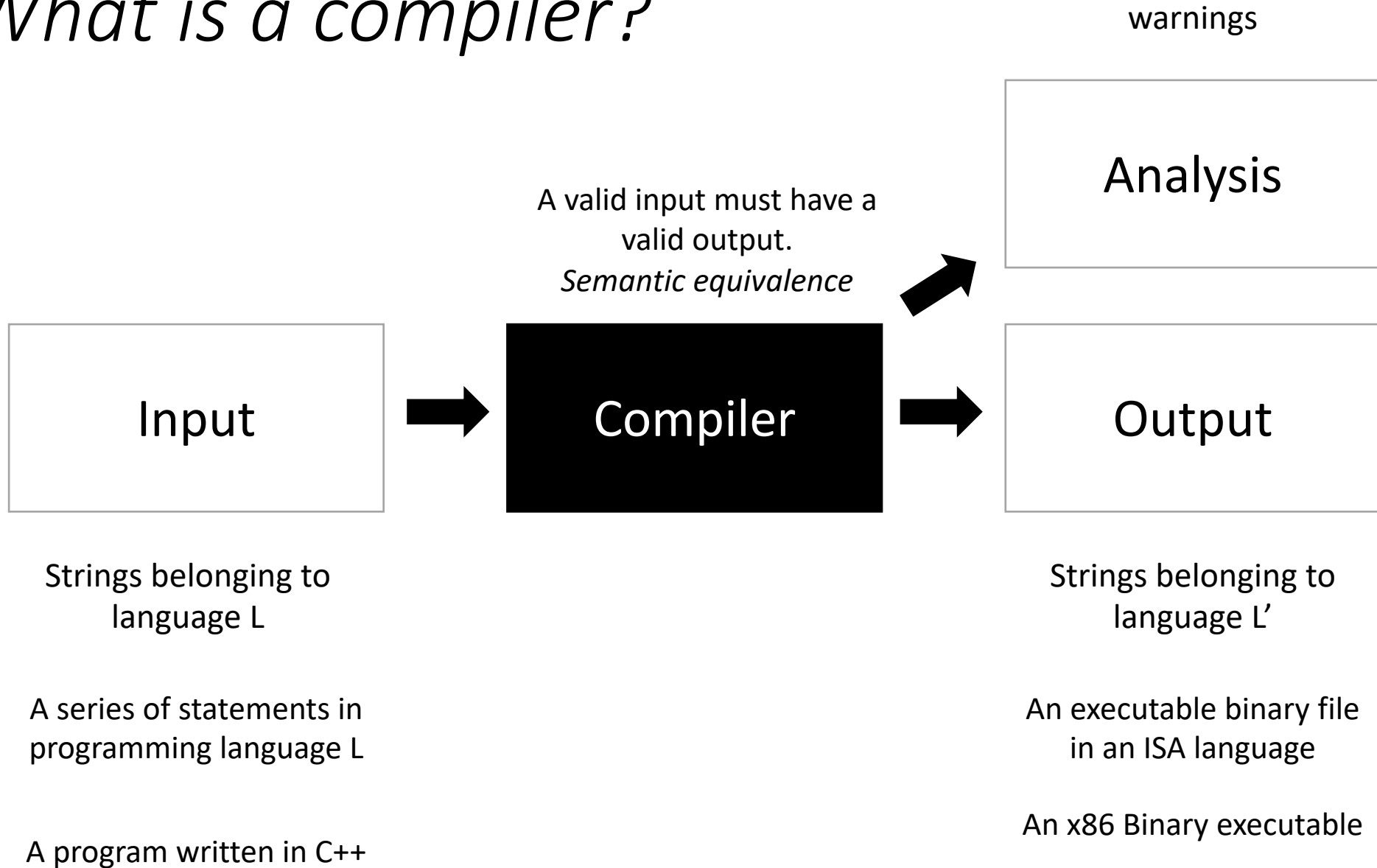
An executable binary file in an ISA language

An x86 Binary executable

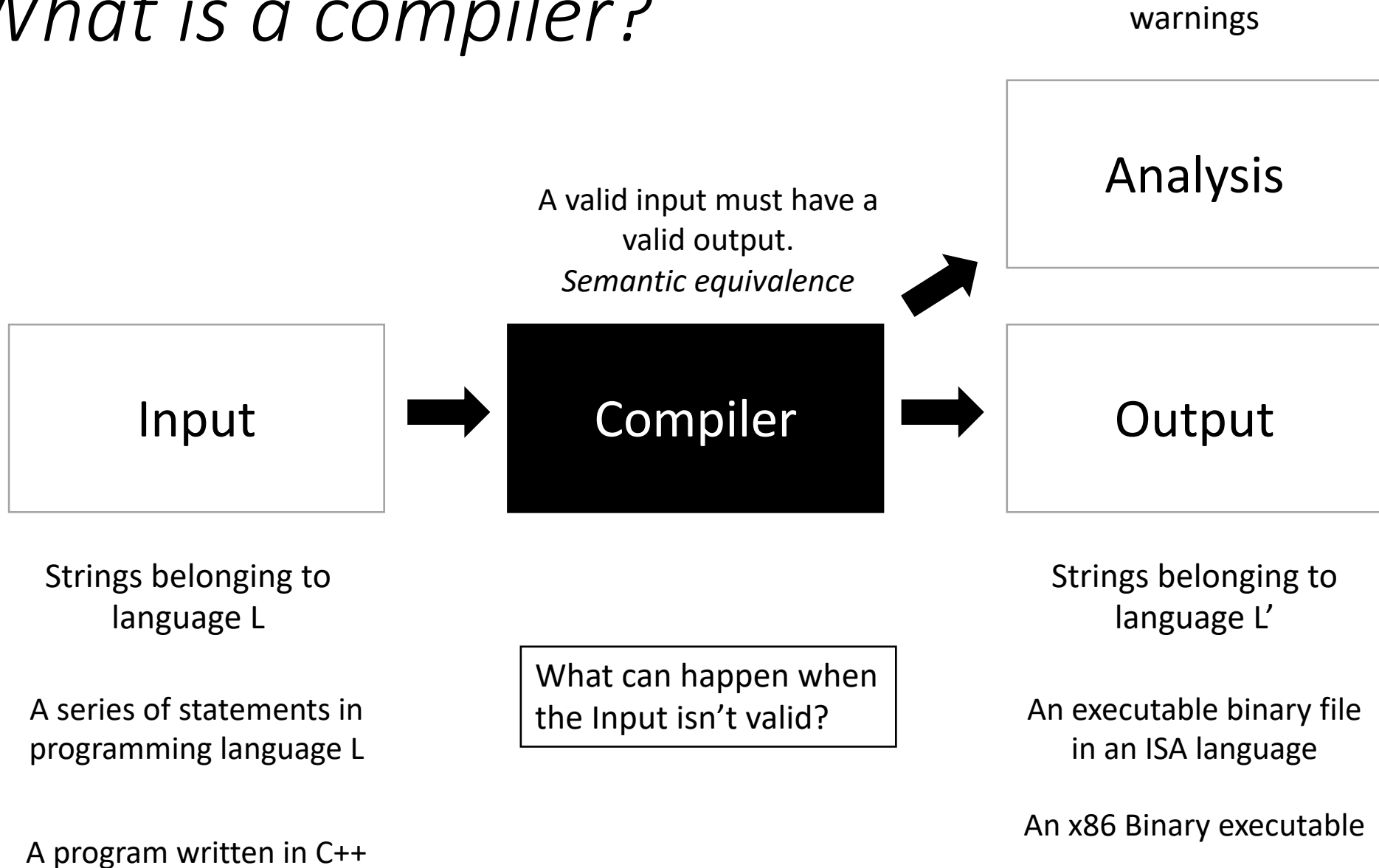
What is a compiler?



What is a compiler?



What is a compiler?



What can happen when the Input isn't valid?

```
int my_var = 5;  
my_var = my_car + 5;
```

Try running this through clang; you will get an error and a suggestion!

What can happen when the Input isn't valid?

```
int my_var = 5;  
my_var = my_car + 5;
```

```
int foo() {  
    int *x = malloc(100*sizeof(int))  
    return x[100];  
}
```

What about this one?

Next class

- Tokenizing with regular expressions