

# Building an Evaluator – Part #2

Michael McThrow

San Jose State University  
Computer Science Department  
CS 152 – Programming Paradigms

September 23, 2020



# Table of Contents

- 1 Recap
- 2 eval and apply
- 3 Preview of Monday's Lecture
- 4 Upcoming Midterm

# Table of Contents

- 1 Recap
- 2 eval and apply
- 3 Preview of Monday's Lecture
- 4 Upcoming Midterm

Let's review environments and function calls in Lecture 9.

# Table of Contents

- 1 Recap
- 2 eval and apply**
- 3 Preview of Monday's Lecture
- 4 Upcoming Midterm

# eval and apply

Scheme and other Lisp languages offer an `eval` function that evaluates any S-expression and a `apply` function that performs a function call given a function and its arguments.

```
(eval '(+ 1 2 3)) ; returns 6
```

```
(apply + '(1 2 3)) ; returns 6
```

# The Power of `eval` and `apply`

The functions `eval` and `apply` make it possible for Scheme programs to arbitrarily execute Scheme expressions that are not part of the source code, which can be very powerful.

# Table of Contents

- 1 Recap
- 2 eval and apply
- 3 Preview of Monday's Lecture**
- 4 Upcoming Midterm



# Monday's Lecture

- We will cover virtual machines and compilation.
- In the context of programming languages, a **virtual machine** is an artificial software-defined computer architecture that is used for hosting a programming language's implementation.
- Programming languages that use a virtual machine compile their code to the virtual machine's *bytecode*, and an interpreter executes the bytecode.
- Java uses the Java Virtual Machine, and Python uses its own virtual machine. Microsoft offers the .NET virtual machine that hosts many programming languages, such as C#, Visual Basic .NET, and F# (a functional programming language).

# Table of Contents

- 1 Recap
- 2 eval and apply
- 3 Preview of Monday's Lecture
- 4 Upcoming Midterm**

# Details

- Midterm will be assigned **Wednesday, October 7, 2020**. I will still host office hours, but there will be no lecture.
- Midterm will be done at home; there is no need to come to campus.
- Midterm will be available through Canvas from midnight to 11:59pm.
- Midterm should take 75 minutes (our normal class period time) to complete, but I will not be enforcing a time limit.
- Midterm must be turned in by 11:59pm Pacific Daylight Time on October 7.
- I can answer *clarifying questions only* via email or Canvas.

# Details

- Midterm will be open book and open note, but books and notes are limited to those that have been assigned in this course.
- Feel free to use your Lab 1, Lab 2, and Project 1 code as notes for the exam.
- Be prepared to write code in Scheme using DrRacket. There may also be a Java coding exercise.
- **DO NOT** collaborate with other people during the exam.
- **DO NOT** share exam questions with others until grades are returned.

# Topics That Are Guaranteed to Be Covered

- Context-free and regular grammars
- Differences between procedural, structured, and functional programming
- Scheme
- Abstract syntax trees
- Evaluation with environments

# Topics That Will NOT Be Covered

- Operational semantics
- Minutes about the syntax of FORTRAN and ALGOL 60
- ANTLR (though expect to see context-free and regular grammars covered)

# Midterm Review Session

- The Monday, October 5 lecture will be a review session for topics that will show up on the midterm.
- I will also be answering questions from students about any of the material covered during the course.