CSCI 334:
Principles of Programming Languages

Lecture 15: Interpretation

Instructor: Dan Barowy

Williams

Outline

Feedback

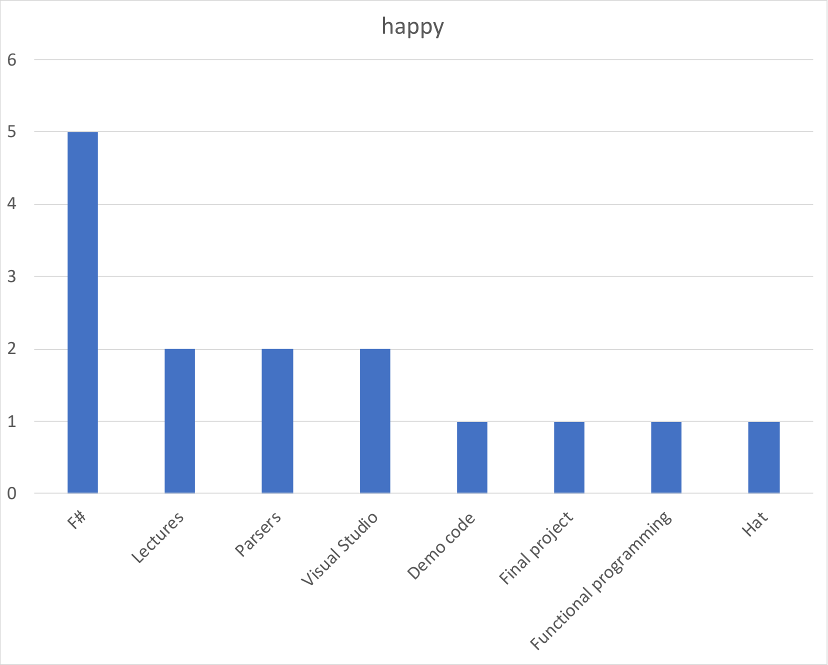
Weekly TODOs

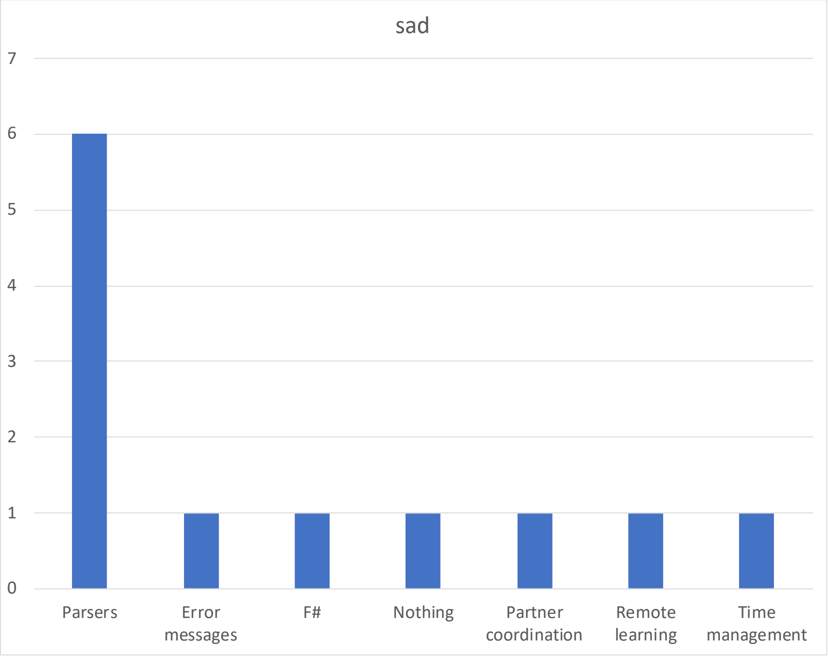
Interpreter Evaluation

Project meetings

Happy to talk to you— reach out!

Feedback



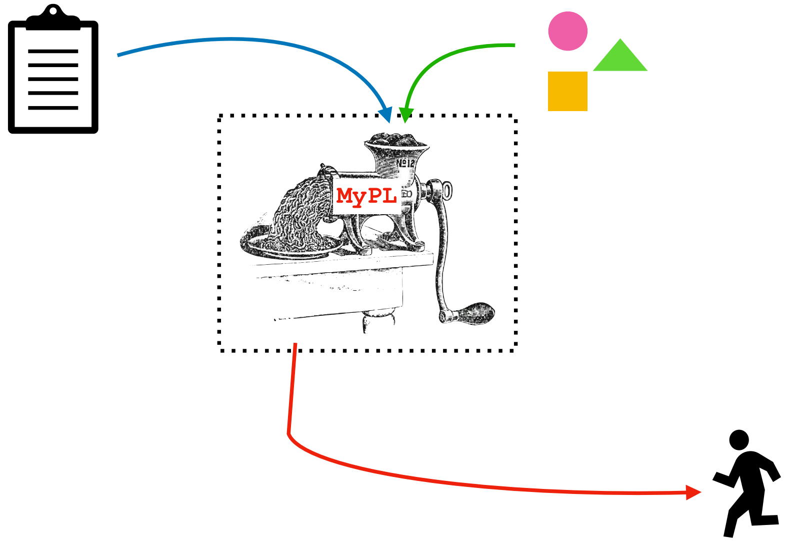


Reminder: Four TODOs each week

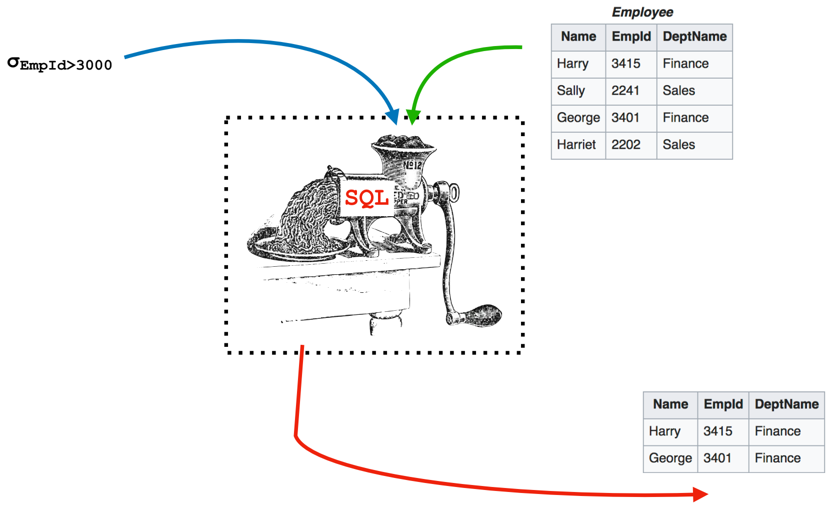
• Lab assignment	5-10 hours
• Quiz	<20 minutes
• Activity	<20 minutes
• Feedback	<5 minutes

What is a programming language?

What is a programming language?



What is a programming language?

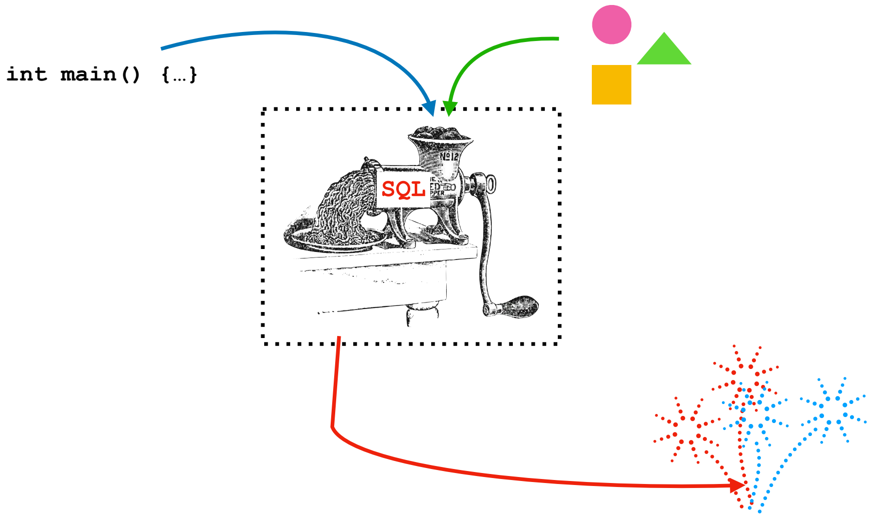

$\sigma_{\text{EmpId}>3000}$

Employee

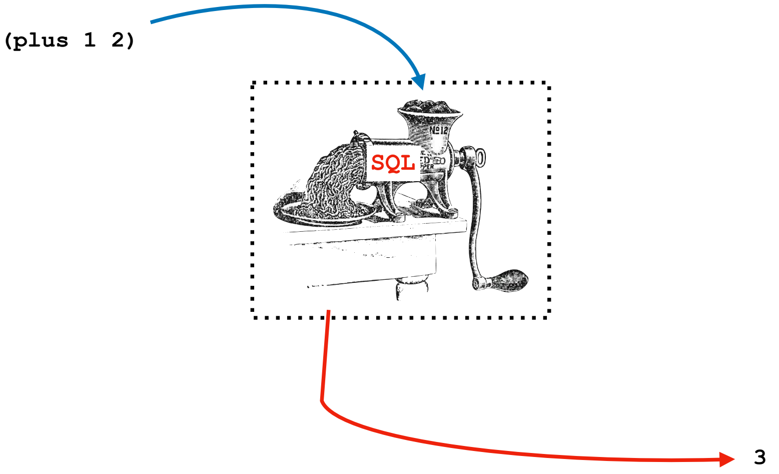
Name	EmpId	DeptName
Harry	3415	Finance
Sally	2241	Sales
George	3401	Finance
Harriet	2202	Sales

Name	EmpId	DeptName
Harry	3415	Finance
George	3401	Finance

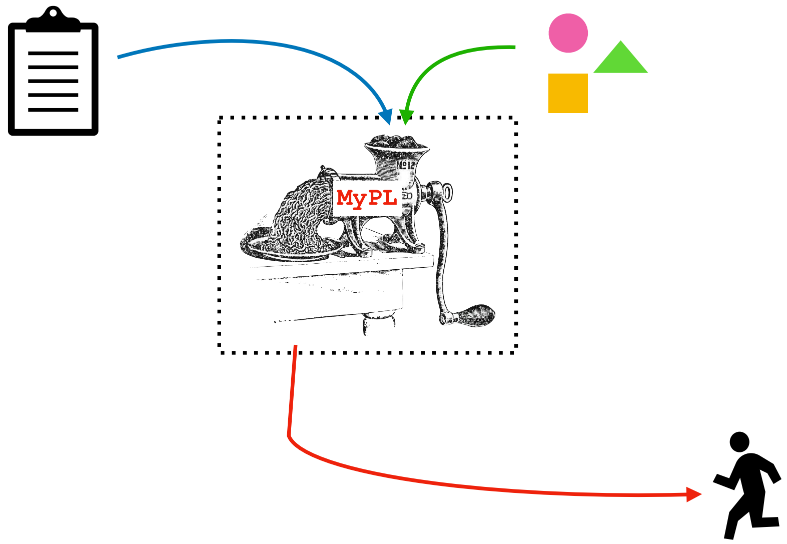
What is a programming language?


```
int main() {…}
```

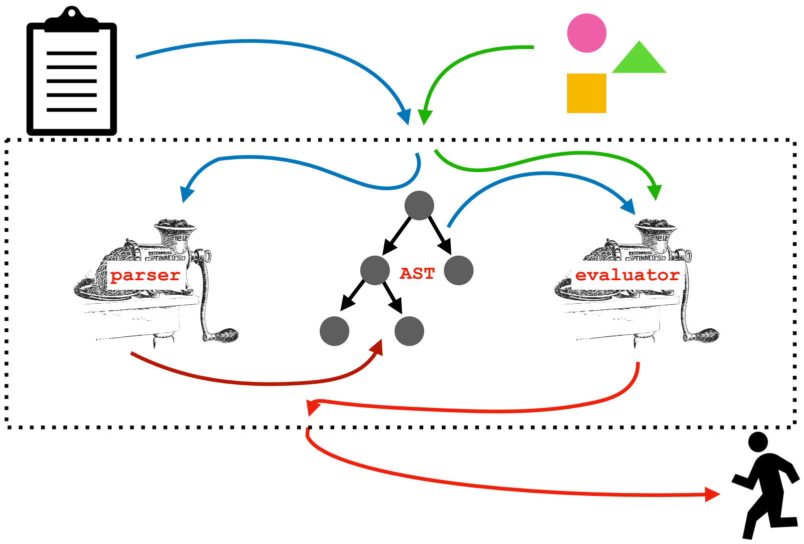
What is a programming language?


```
(plus 1 2)
```

What is a programming language?



What is a programming language?



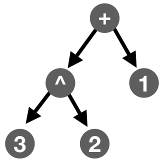
Program Interpreter

Program Interpreter

A **program interpreter** is a computer program that "interprets" given statements or expressions in a programming language. Unlike a compiler, an interpreter **directly** interprets code, often in the form of an **abstract syntax tree**.

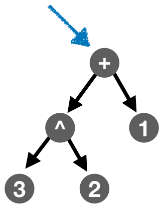
Example

```
3^2 + 1
```



Example

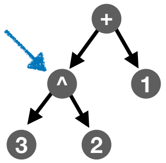
```
3^2 + 1
```



Eager evaluation: usually a **post-order traversal** of an AST.

Example

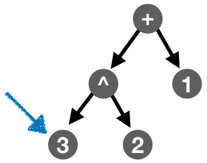
```
3^2 + 1
```



Eager evaluation: usually a **post-order traversal** of an AST.

Example

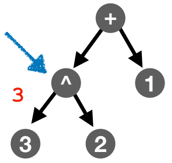
```
3^2 + 1
```



Eager evaluation: usually a **post-order traversal** of an AST.

Example

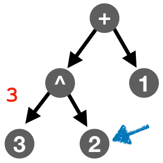
`3^2 + 1`



Eager evaluation: usually a **post-order traversal** of an AST.

Example

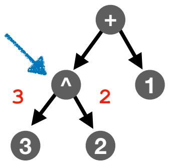
`3^2 + 1`



Eager evaluation: usually a **post-order traversal** of an AST.

Example

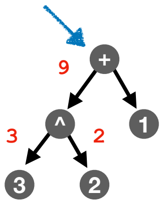
`3^2 + 1`



Eager evaluation: usually a **post-order traversal** of an AST.

Example

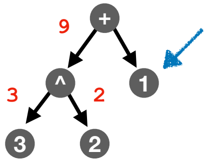
`3^2 + 1`



Eager evaluation: usually a **post-order traversal** of an AST.

Example

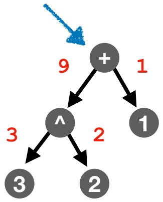
```
3^2 + 1
```



Eager evaluation: usually a **post-order traversal** of an AST.

Example

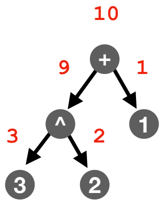
```
3^2 + 1
```



Eager evaluation: usually a **post-order traversal** of an AST.

Example

```
3^2 + 1
```



Eager evaluation: usually a **post-order traversal** of an AST.

This traversal is conveniently written as a **recursive function**.

Code Example

pluslang

```
<expr> ::= (plus <expr> <expr>+)
        | n∈ℕ
```

```
(plus 1 2)

(plus 1 2 3 4 5)
```

(code)

Variables

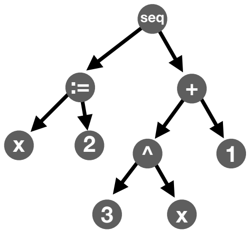
Variables

A **variable** is a named placeholder for a value in an expression. At runtime, when a value is **assigned** to a variable, that **variable name is bound to the value** within the variable's scope. When a variable is **used** in an expression, the bound value is **substituted** into the expression when the expression is evaluated.

We'll talk about **scope** in the next lecture.

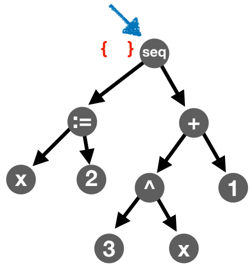
Example

```
x := 2
3^x + 1
```



Example

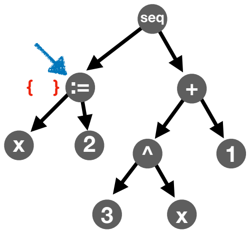
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

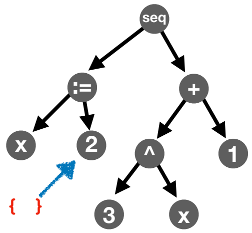
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

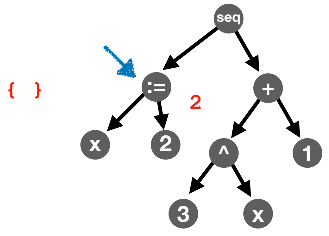
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

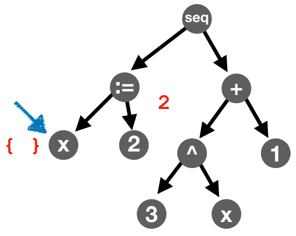
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

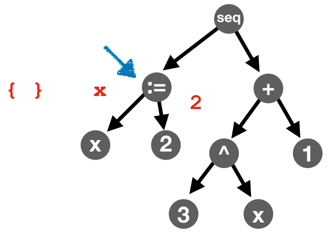
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

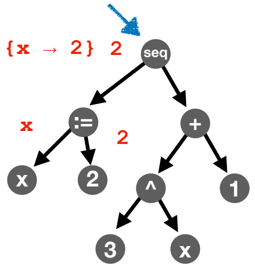
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

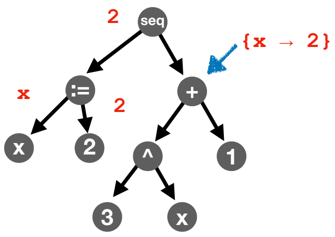
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

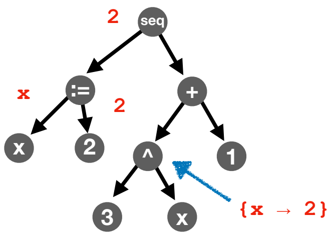
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

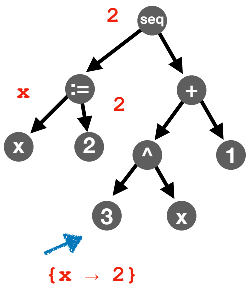
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

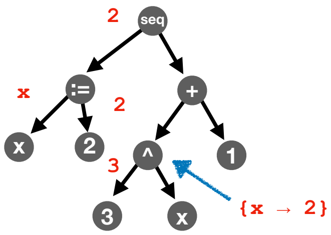
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

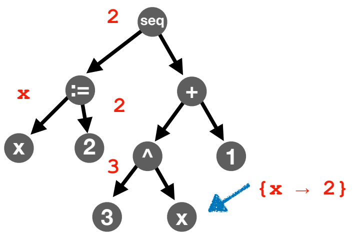
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

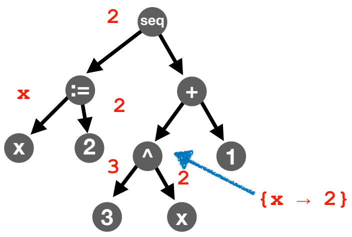
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

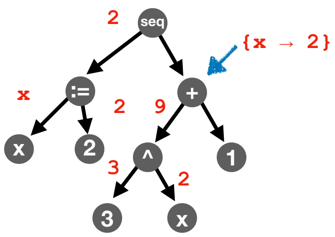
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

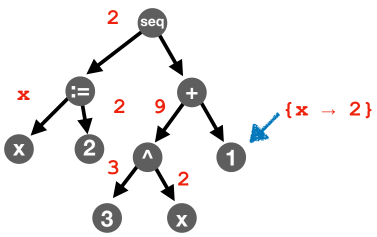
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

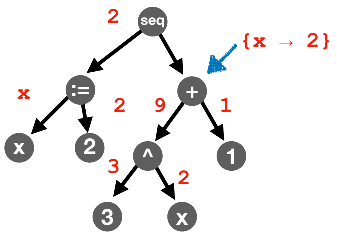
```
x := 2
3^x + 1
```



{ } is an “environment”

Example

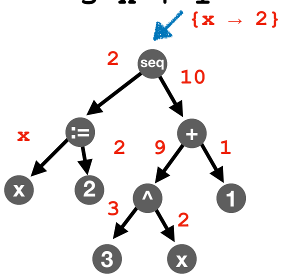
```
x := 2
3^x + 1
```



{ } is an "environment"

Example

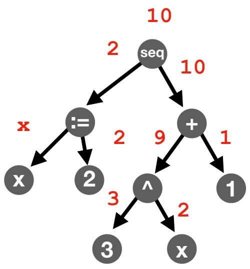
```
x := 2
3^x + 1
```



{ } is an "environment"

Example

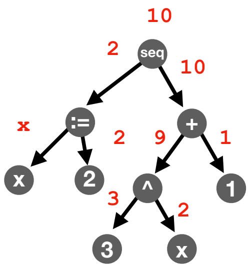
```
x := 2
3^x + 1
```



{ } is an "environment"

Example

```
x := 2
3^x + 1
```



Cool, huh?

Every CS major should know this.

Recap & Next Class

This lecture:

Program Interpretation

Next lecture:

Scope

Growing a Language