Programming Paradigms
Names, Scopes, and Bindings (Part 1)

Prof. Dr. Michael Pradel
Software Lab, University of Stuttgart
Summer 2020
Names in PLs

Abstraction in two dimensions

■ From hardware
  □ Variable names abstract away how exactly values are stored

■ From implemented functionality
  □ Function names abstract from the implemented behavior
Binding

- **Association between entities and their names**, e.g.,
  - A variable bound to a memory object
  - A function bound to the code implementing the function
- **Different languages have different rules**
  - E.g., static vs. dynamic binding
Scope

- **Scope of a binding**: Textual region where binding is active
- **Scope**: Maximal region where no bindings change

Example (Python):

```python
x = 1
def f():
    x = 2
    y = x
```
Scope

- **Scope of a binding**: Textual region where binding is active
- **Scope**: Maximal region where no bindings change

Example (Python):

```python
x = 1

def f():
    x = 2
    y = x

Outer scope

Scope of function
```
Overview

- Object lifetime and storage management
- Scopes
- Aliasing and overloading
- Binding of referencing environments