

# Names, Scopes, Bindings

## Textbook Chapter 3

1

---

---

---

---

---

---

---

---

# Binding

- Association of name with value
  - language design time operators
  - language implementation time size of int in C++
  - program writing time
  - compile time +
  - link time
  - load time
  - run time virtual functions
- Static vs. dynamic

2

---

---

---

---

---

---

---

---

# Object Lifetimes

- Creating of object
- Creation of binding
- Use of binding
- Deactivation/reactivation of binding
- Destruction of binding
- Destruction of object

3

---

---

---

---

---

---

---

---

## Storage Allocation

- **Static**
  - allocated at link time      **static fields**
- **Stack**
  - allocated in function call      **local vars**
- **Heap**
  - allocated dynamically      **using new**

4

---

---

---

---

---

---

---

---

## Static vs. Dynamic Scope

```
(define x 1)
(define (bar) x)
(define (foo f)
  (define x 2)
  (f))
(foo bar)
```

- Static scoping:      **x = 1**
- Dynamic scoping:      **x = 2**

5

---

---

---

---

---

---

---

---

## Implementation of Static Scoping

- Use Closures as function values
  - Pointer to code
  - Pointer to environment
- Remember environment in which function is defined

6

---

---

---

---

---

---

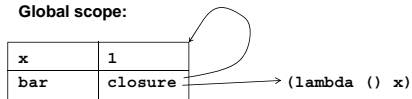
---

---

## Construction of Closures

```
(define x 1)
(define (bar) x)
```

Global scope:



Closure contains environment in which function is defined

7

---

---

---

---

---

---

---

---

## Function Call

- Take environment out of closure
- Create function scope
- Define parameters in function scope
- Evaluate function body in fun scope

8

---

---

---

---

---

---

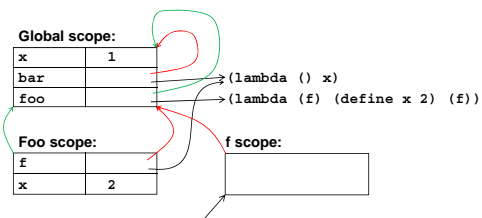
---

---

## Construction of Function Scopes

```
(define x 1)
(define (bar) x)
(define (foo f) (define x 2) (f))
(foo bar)
```

Global scope:



9

---

---

---

---

---

---

---

---

## Functions as Return Values

```
(define (add x)
  (lambda (y)
    (+ x y)))

(define add1 (add 1))
(define add5 (add 5))
(define i (add1 7))
(define j (add5 7))
```

10

---

---

---

---

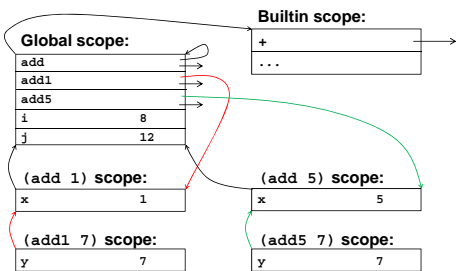
---

---

---

---

## Environments for Add Example



11

---

---

---

---

---

---

---

---