



Swift Labs: Breaking Cycles

• cyclic references cause memory leaks

- References in Swift are strong by default

• To break a weak reference used

• There are n i led when the last strong reference is destroyed

• Must be checked before dereferencing

7

6


```
var p = Person()  
p.set_first_name (new_first: "Dave")  
print (p.get_first_name())  
  
weak var p2 = p  
p2?.set_first_name (new_first: "John")
```



Swift **class**: Breaking Cycles

- Cyclic references cause memory leaks
- References in Swift are **strong** by default
- To break a cycle **weak** references can be used
- These are **nil**ed when the last strong reference is destroyed
- Must be checked before dereferencing

```
var p = Person()  
p.set_first_name (new_first: "Dave")  
print (p.get_first_name())  
  
weak var p2 = p  
p2?.set_first_name (new_first: "John")
```



Wrapped `std::weak_ptr`