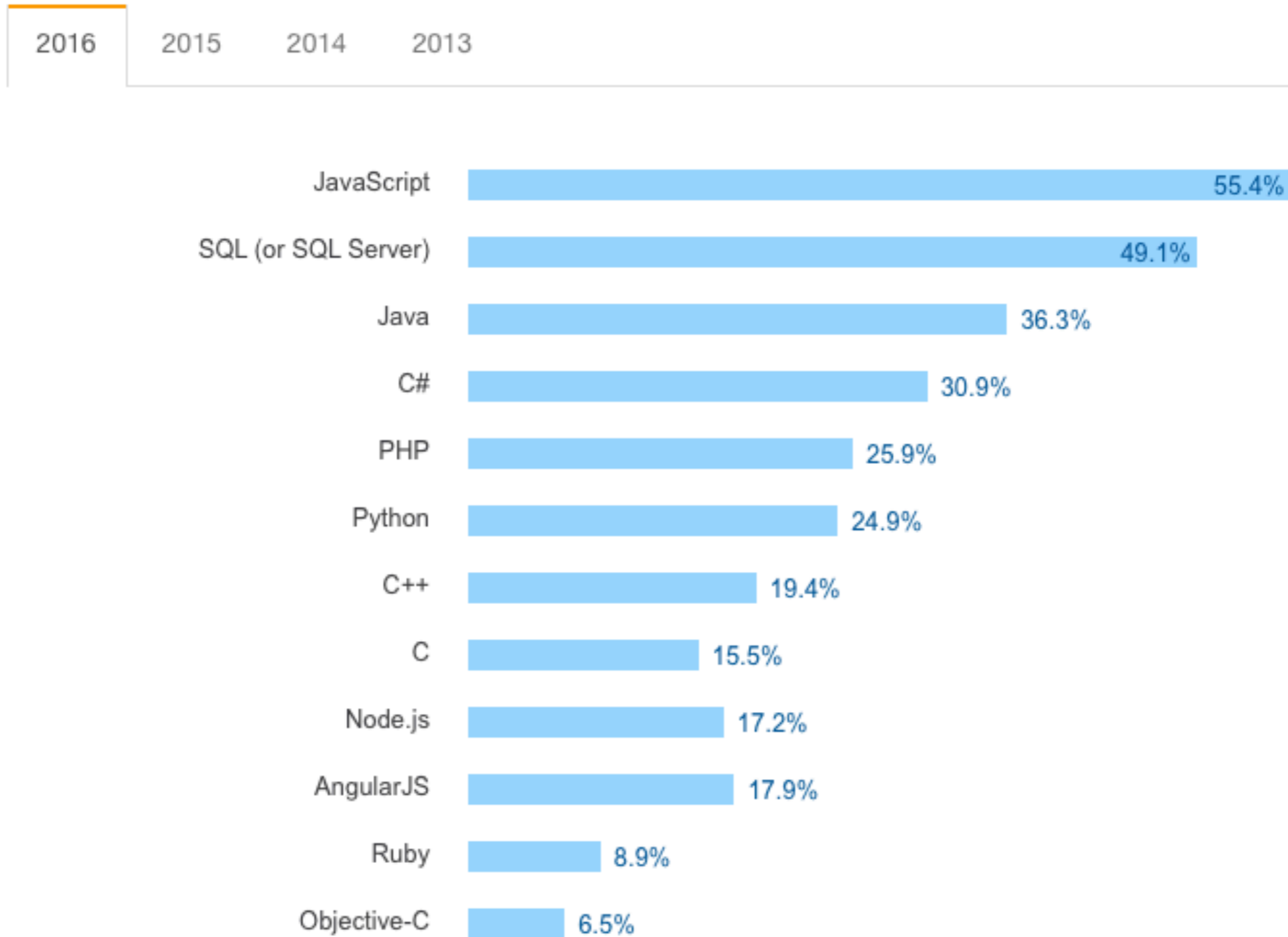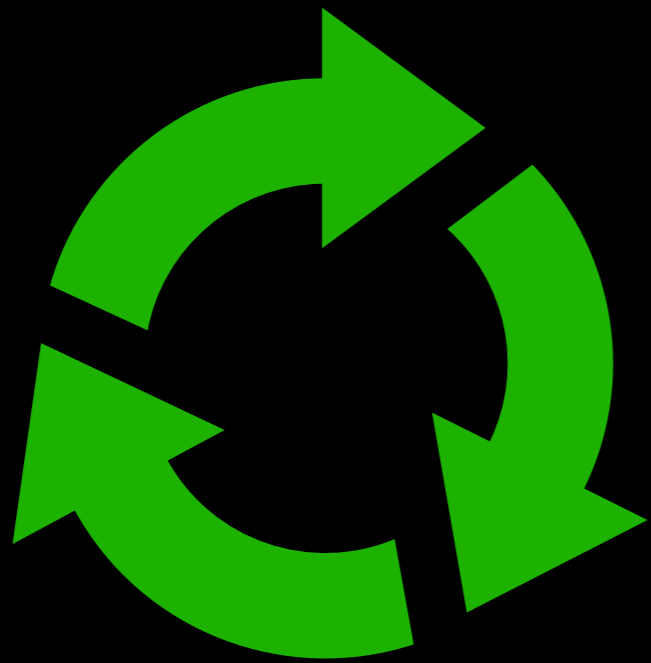# #5 Object Oriented Programming
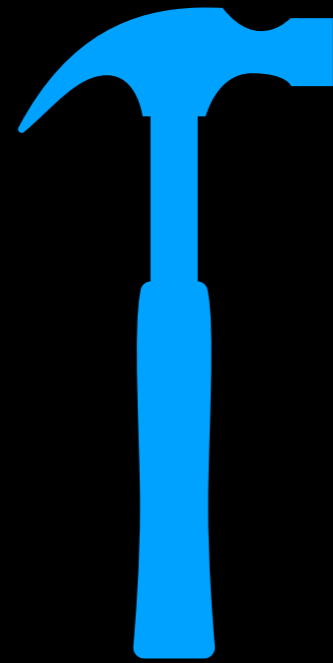
TA: Jerry Chen (jerry.c@berkeley.edu)

Did you hear about the new object oriented get-rich-quick scheme? It's called "inheritance."

I. Most Popular Technologies

Technology	Percentage
JavaScript	55.4%
SQL (or SQL Server)	49.1%
Java	36.3%
C#	30.9%
PHP	25.9%
Python	24.9%
C++	19.4%
C	15.5%
Node.js	17.2%
AngularJS	17.9%
Ruby	8.9%
Objective-C	6.5%

49,397 responses

OOP in Python

Some vocabulary

- Class - a "template" for a object

- Object - an instance of a class

- Attribute - data

  - **Instance attributes** are specific to an object

  - **Class attributes** are shared with objects of the class

- Method - a function that has been bound to an object

```python
class Instructor:
    degree = "PhD (Magic)"
    def __init__(self, name):
        self.name = name
    def lecture(self, topic):
        print("Today we're learning about " + topic)
```

```python
class Instructor:
    degree = "PhD (Magic)"
    def __init__(self, name):
        self.name = name
    def lecture(self, topic):
        print("Today we're learning about " + topic)

dumbledore = Instructor("Dumbledore")
```

```python
1 class Instructor:
2     degree = "PhD (Magic)"
3     def __init__(self, name):
4         self.name = name
5     def lecture(self, topic):
6         print("Today we're learning about " + topic)
7
8 dumbledore = Instructor("Dumbledore")
```

```python
class Instructor:
    degree = "PhD (Magic)"
    def __init__(self, name):
        self.name = name
    def lecture(self, topic):
        print("Today we're learning about " + topic)

dumbledore = Instructor("Dumbledore")
```

```python
1 class Instructor:
2     degree = "PhD (Magic)"
3     def __init__(self, name):
4         self.name = name
5     def lecture(self, topic):
6         print("Today we're learning about " + topic)
7
8 dumbledore = Instructor("Dumbledore")
9 dumbledore.lecture("Magic")
```

```python
1 class Instructor:
2     degree = "PhD (Magic)"
3     def __init__(self, name):
4         self.name = name
5     def lecture(self, topic):
6         print("Today we're learning about " + topic)
7
8 dumbledore = Instructor("Dumbledore")
9 dumbledore.lecture("Magic")
```

```python
class Car:
    def drive(self):
        print("I am definitely a car")

class Boat:
    def __init__(self):
        self.is_car = 'Nope'
b = Boat()

Car.drive(b)
b.drive()
Car.drive("car")
Car.drive()
```

```python
class Car:
    def drive(self):
        print("I am definitely a car")

class Boat:
    def __init__(self):
        self.is_car = 'Nope'
b = Boat()

Car.drive(b)          ✅
b.drive()
Car.drive("car")
Car.drive()
```

```python
class Car:
    def drive(self):
        print("I am definitely a car")

class Boat:
    def __init__(self):
        self.is_car = 'Nope'
b = Boat()

Car.drive(b)        ✅
b.drive()           ❌
Car.drive("car")
Car.drive()
```

```python
class Car:
    def drive(self):
        print("I am definitely a car")

class Boat:
    def __init__(self):
        self.is_car = 'Nope'
b = Boat()

Car.drive(b)        ✔
b.drive()           ✖
Car.drive("car")    ✔
Car.drive()
```

```python
class Car:
    def drive(self):
        print("I am definitely a car")

class Boat:
    def __init__(self):
        self.is_car = 'Nope'
b = Boat()

Car.drive(b)        ✅
b.drive()           ❌
Car.drive("car")    ✅
Car.drive()         ❌
```

```python
class Car:
    def __init__(not_self):
        not_self.tires = 10

class Funky:
    def __init__():
        print("No self?")

class BoatCar(Boat):
    def drive():
        print("I am definitely... a boatcar")

b = BoatCar()
b.drive()
BoatCar.drive()
```

```python
class Car:
    def __init__(not_self):
        not_self.tires = 10

class Funky:
    def __init__():
        print("No self?")

class BoatCar(Boat):
    def drive():
        print("I am definitely... a boatcar")

b = BoatCar()
b.drive()
BoatCar.drive()
```

```python
class Car:
    def __init__(not_self):
        not_self.tires = 10

class Funky:
    def __init__():
        print("No self?")

class BoatCar(Boat):
    def drive():
        print("I am definitely... a boatcar")

b = BoatCar()
b.drive()
BoatCar.drive()
```

```python
class Car:
    def __init__(not_self):          ✓
        not_self.tires = 10

class Funky:
    def __init__():                  ✗ ✓
        print("No self?")

class BoatCar(Boat):
    def drive():
        print("I am definitely... a boatcar")

b = BoatCar()
b.drive()
BoatCar.drive()
```

```python
class Car:
    def __init__(not_self):
        not_self.tires = 10

class Funky:
    def __init__():
        print("No self?")

class BoatCar(Boat):
    def drive():
        print("I am definitely... a boatcar")

b = BoatCar()
b.drive()
BoatCar.drive()
```

```python
class Car:
    def __init__(not_self):      ✓
        not_self.tires = 10

class Funky:
    def __init__():              ✗
        print("No self?")

class BoatCar(Boat):             ✓
    def drive():
        print("I am definitely... a boatcar")

b = BoatCar()
b.drive()
BoatCar.drive()
```

```
 1 class Car:
 2     def __init__(not_self):        ✓
 3         not_self.tires = 10
 4
 5 class Funky:
 6     def __init__():                ✗
 7         print("No self?")
 8
 9 class BoatCar(Boat):               ✓
10     def drive():
11         print("I am definitely... a boatcar")
12
13 b = BoatCar()        ✓
14 b.drive()
15 BoatCar.drive()
```

```python
1  class Car:
2      def __init__(not_self):        ✅
3          not_self.tires = 10
4
5  class Funky:
6      def __init__():                ❌
7          print("No self?")
8
9  class BoatCar(Boat):               ✅
10     def drive():
11         print("I am definitely... a boatcar")
12
13 b = BoatCar()      ✅
14 b.drive()          ❌
15 BoatCar.drive()
```

```python
1  class Car:
2      def __init__(not_self):          ✅
3          not_self.tires = 10
4
5  class Funky:
6      def __init__():                  ❌
7          print("No self?")
8
9  class BoatCar(Boat):                 ✅
10     def drive():
11         print("I am definitely... a boatcar")
12
13 b = BoatCar()          ✅
14 b.drive()              ❌
15 BoatCar.drive()        ✅
```