

# C and C++

## VI

Spring 2014  
Carola Wenk

# Iterators in Python, Java, C++

- In Python, the “in” keyword provided a way to iterate through many different kinds of sequences/collections.

```
L = [1, 2, 3, 'abc', 1.0, 'foo']  
  
for i in L:  
    print i
```

# Iterators in Python, Java, C++

- In Python, the “in” keyword provided a way to iterate through many different kinds of sequences/collections.

```
L = [1, 2, 3, 'abc', 1.0, 'foo']

for i in L:
    print i

D = {'a':2, 'b':3, 'c':4}

# what order?
for i in D:
    print i
```

- Can we do this in Java or C++? How can “iteration” be defined for any type where it makes sense?

# Iterators in Python, Java, C++

- Can we do this in Java or C++? How can “iteration” be defined for any type where it makes sense?

```
...
ArrayList<String> A = new ArrayList<String>();

// fill up A
...

Iterator itr = A.iterator();
while (itr.hasNext()) {
    System.out.println(itr.next());
}
```

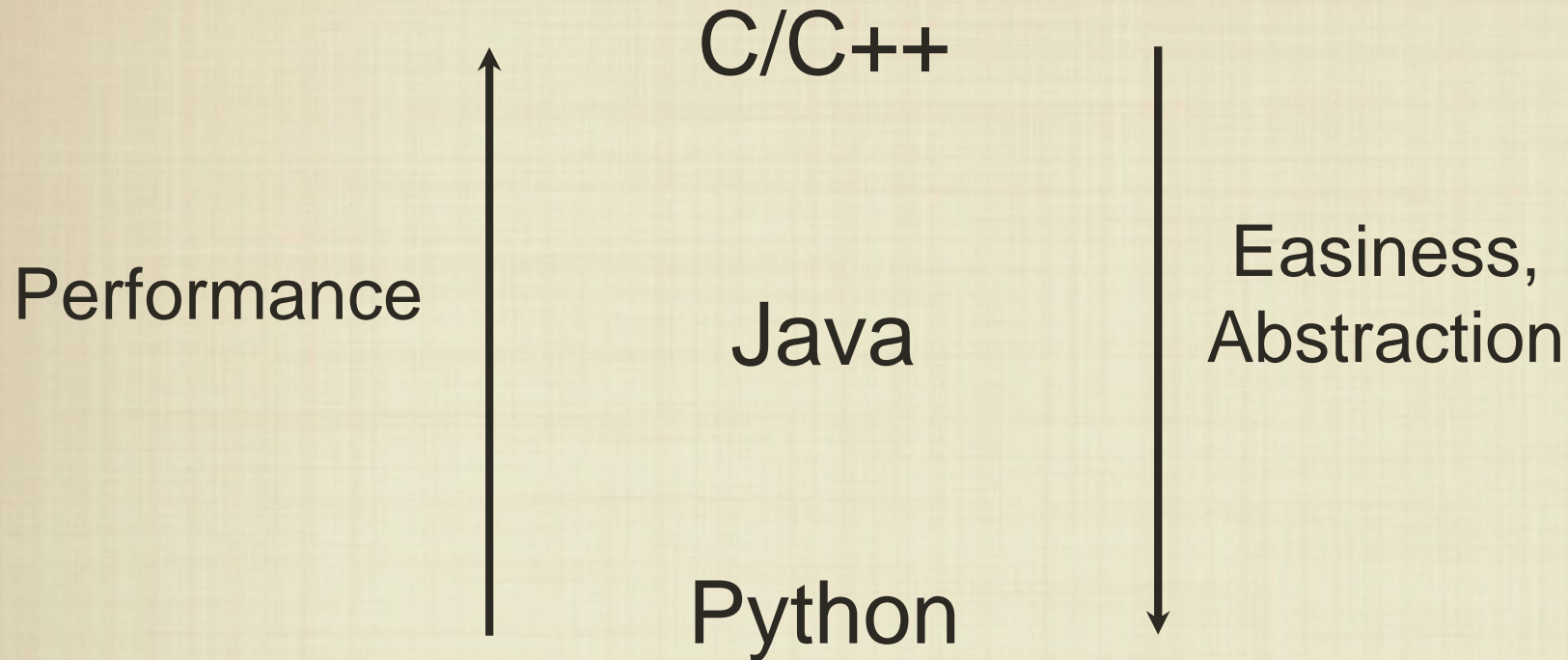
# Iterators in Python, Java, C++

- Can we do this in Java or C++? How can “iteration” be defined for any type where it makes sense?

```
...  
vector<string> V;  
  
// fill up V  
...  
  
for (std::vector<string>::iterator it = V.begin();  
     it != V.end(); ++it) {  
    cout << *it << endl;  
}
```

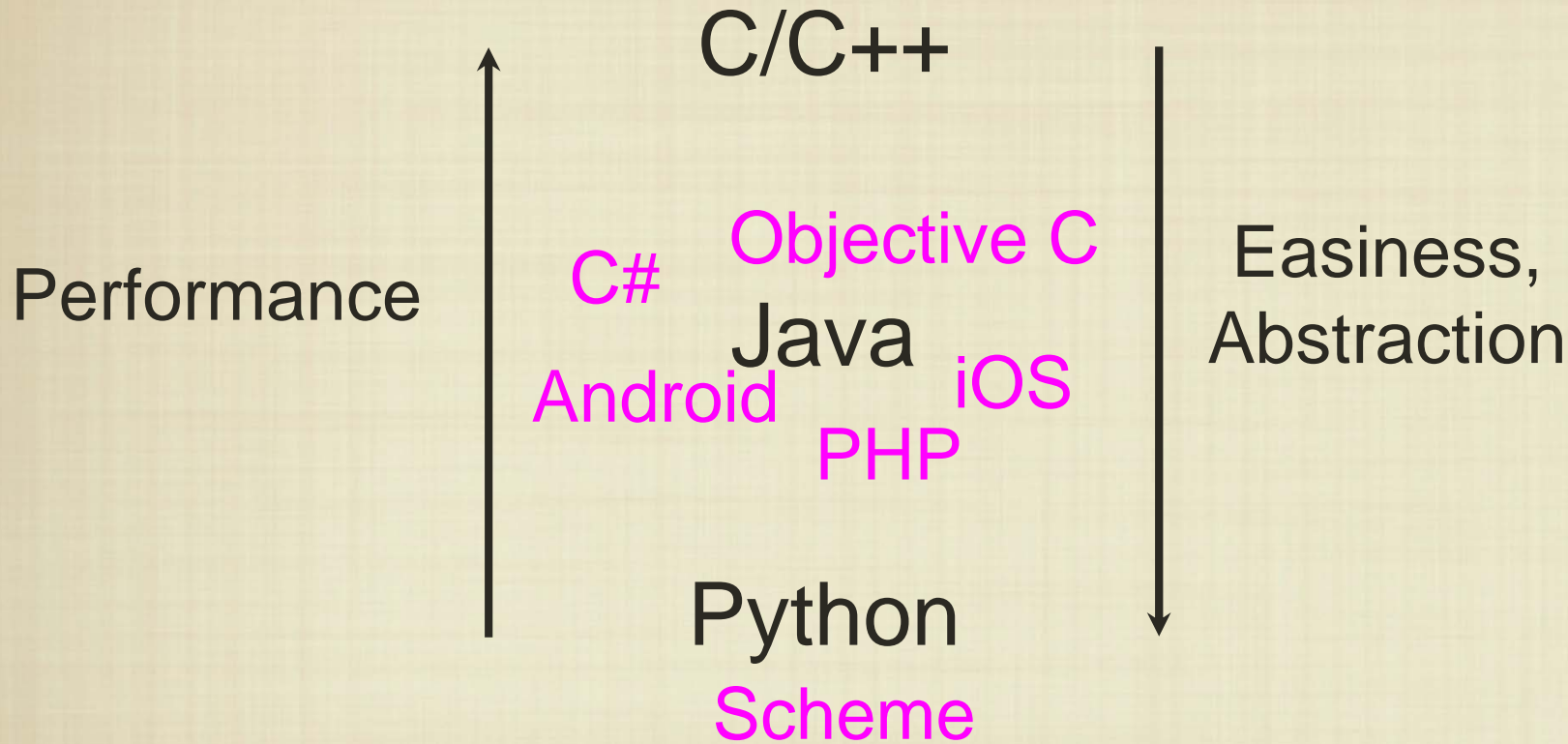


# Bottom Line



- The typical rule of thumb is to use the “easiest” language that will suit your needs.

# Bottom Line



- The typical rule of thumb is to use the “easiest” language that will suit your needs.