

Polymorphism

What is polymorphism?

In English, *polymorphism* means having many forms.

- In biology, polymorphism closely related to biodiversity, genetic variation and adaptation.
- In computing, polymorphism refers to the notion of a single expression having multiple types.

December 7, 2010

2

Simple example

Since Python is weakly-typed (unlike Java), polymorphism is implicit and everywhere:

```
x = "Trinity"  
print x  
x = 1823  
print x
```

Here, the type of `x` in `print x` is not explicit and thus has *multiple* possibilities.

December 7, 2010

3

Sorting example

```
def sort(list):
    for i in range(len(list)):
        minIndex = i
        for j in range(i + 1, len(list)):
            if list[j] < list[minIndex]:
                minIndex = j
        if minIndex != i:
            swap(list, i, minIndex)
```

```
L = [4, 1, 5, 9, 3, 2, 6, 7]
sort(L)
print L
```

December 7, 2010

4

Sorting example

```
def sort(list):
    for i in range(len(list)):
        minIndex = i
        for j in range(i + 1, len(list)):
            if list[j] < list[minIndex]:
                minIndex = j
        if minIndex != i:
            swap(list, i, minIndex)
```

```
L = ["red", "fun", "hot", ..., "interesting"]
sort(L)
print L
```

December 7, 2010

5

Polymorphic <

The function `sort` is able to sort lists of any type because `<` is *polymorphic*; that is, the same `<` operator can be used to compare integers, words, etc...

This is possible because Python supports operator overloading. Specifically, `<` can be defined for comparing any objects via the method `__cmp__`.

December 7, 2010

6

Sorting example

```
def sort(list):
    for i in range(len(list)):
        minIndex = i
        for j in range(i + 1, len(list)):
            if list[j] < list[minIndex]:
                minIndex = j
        if minIndex != i:
            swap(list, i, minIndex)
```

An appropriate `__cmp__` method is invoked for `<` based on what is inside `list`.

December 7, 2010

7

How about Java?

```
public void sort(int A[]) {
    for (int i = 0; i < A.length - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < A.length; j++)
            if (A[j] < A[minIndex])
                minIndex = j;
        if (minIndex != i)
            swap(A, i, minIndex);
    }
}
```

December 7, 2010

8

How about Java?

```
public void sort(int A[]) {
    for (int i = 0; i < A.length - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < A.length; j++)
            if (A[j] < A[minIndex])
                minIndex = j;
        if (minIndex != i)
            swap(A, i, minIndex);
    }
}
```

This only works for `int` arrays.

December 7, 2010

9

How about Java?

```
public void sort(int A[]) {
    for (int i = 0; i < A.length - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < A.length; j++)
            if (A[j] < A[minIndex])
                minIndex = j;
        if (minIndex != i)
            swap(A, i, minIndex);
    }
}
```

In Java, < cannot be overloaded.

December 7, 2010

10

How about Java?

```
public void sort(Comparable A[]) {
    for (int i = 0; i < A.length - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < A.length; j++)
            if (A[j].compareTo(A[minIndex]) < 0)
                minIndex = j;
        if (minIndex != i)
            swap(A, i, minIndex);
    }
}
```

Instead, we use `compareTo()`.

December 7, 2010

11

Comparable

- Java's `compareTo()` is analogous to Python's `__cmp__`.
- Many Java classes (such as `String`) support `compareTo()`.
- You can also define your own Java class with your own implementation of `compareTo()`.
- Any Java class that supports `compareTo()` is called `Comparable`.

December 7, 2010

12

How about Java?

```
public void sort(Comparable A[]) {  
    for (int i = 0; i < A.length - 1; i++) {  
        int minIndex = i;  
        for (int j = i + 1; j < A.length; j++)  
            if (A[j].compareTo(A[minIndex]) < 0)  
                minIndex = j;  
        if (minIndex != i)  
            swap(A, i, minIndex);  
    }  
}
```

Here, A is an array of **Comparable** objects.

December 7, 2010

13

How about Java?

Here are some examples of sorting
Comparable objects in Java.

```
Integer N[] = {4, 1, 5, 9, 3, 2, 6, 7};  
String S[] = {"red", "fun", ..., "interesting"};  
sort(N);  
sort(S);
```

December 7, 2010

14
