UNIVERSITY of WISCONSIN
LA CROSSE
COMPUTER SCIENCE

CS 224 Introduction to Python

List Comprehension

1

What is a list comprehension?

- Concise way to create a list from another list
  - Syntax:
    - `L2 = [expression(i) for i in L1 if condition(i)]`

```
cubes = []
for i in range(10):
    cubes.append(i**3)
```

```
cubes = [i**3 for i in range(10)]
```

Equivalent

2

Example using files

The existing list in a comprehension can be ad hoc:

```
path = my_dir
files = [path + '/' + i for i in os.listdir(path)]

for f in files:
    fn = open(f, 'r')
    do something with fn
    fn.close()
```

returns a (Python) list of files

What does this code do?

elements in that list are used to create a new list in which each element also includes the path

3

Filtering

We can choose which elements in the existing list are used to create the new list:

```
roots = [math.sqrt(i) for i in nums if i >= 0]
```

avoids math domain error (sqrt of negative number)

Note: `len(roots) <= len(nums)`

4

Multiple Lists

You can use multiple existing lists to create the new list:

```
pairs = [(x, y) for x in xcoords for y in ycoords]
```

What are the elements of `pairs`?

- each is an (x, y) pair in the cross-product of xcoords and ycoords
- this works even if the two input lists have different lengths

What if you want elements paired by position?

```
pairs = [(xcoords[i], ycoords[i]) for i in
                                    range(len(xcoords))]
```

Can use min if the input lists have different lengths.

5

Using a function

Let coords be a list containing 2-element lists of GPS coordinates:
[[(lat1, lon1), (lat2, lon2)], [(lat3, lon3), (lat4, lon4)] …]

Create a list of distances between the pair of cities in each sublist.

```
def distance(city1, city2):
    # compute and return distance

dists = [distance(x, y) for x, y in coords]
```

Function definition

6

More filtering

Let `data` be a list containing lists of instrument readings:
[[d0_0, d0_1, … d0_n], [d1_0, d1_1, … d1_m] …]

Create a list of the min readings for each sublist.

`min` is a built-in function

```
mins = [min(L) for L in data]
```

Above gives error if a sublist is empty – can't min([])

```
mins = [min(L) for L in data if len(L) > 0]
```

7

"Double iteration"

Let data be a list containing lists of instrument readings:
[[d0_0, d0_1, … d0_n], [d1_0, d1_1, … d1_m] …]

Combine all of the elements into a single list.

```
all_data = [x for L in data for x in L]
```

If we only want non-negative values:

```
all_data = [x for L in data for x in L if x >= 0]
```

8