UW CSE 190p Section

8/9, Summer 2012

Dun-Yu Hsiao

Outlines

- Data Abstraction
- Course Review
- Questions

Data Abstraction

- Representing the essential features of something without including background or inessential detail.

- The **process** of deciding which parts of the implementation that should be **hidden**.

Ex1-Plotting Points

- Given points data ('euclidean', x, y) in a text file, you want to plot the points:
  - Save each point as a tuple
  - Make all the points as a list
  - Print all the points
  - Plot them

- What are the necessary functions?

Plotting Points

- Now the supplied input file changes its format, (‘euclidean’, x, y) → (‘polar’, r, theta)

- Which part of the code will break?

Plotting Points

- Modify your code to accept the new type of data.

- Note how many parts you need to change!

Plotting Points

- Now says your clients take the point data and implement the plot function by themselves.
- You don’t want to bother them every time the input format is changed.
- Can we reduce the amount of modification?
- Evaluate the necessary components of each function.
- What data can be abstract to concept?

Plotting Points

- Now the data format changes again: ('polar', r, theta) → (color, 'polar', r, theta)
- Modify your code to accept the new type of data.
- Note how many parts you need to change!

Ex2-Fraction Operation

- Input: tuples of (n,d)
- You want to implement some basic fraction operation such as plus, product, and equal.

- How would you plan the data and function to increase maintainability?

Course Review

- Manipulation with basic data types:
  - List, set, dictionary, number
- List slice and comprehension
- Integer/floating point calculation
- If, for, function
- English to code
- Sequence of expression evaluation
- Visualize stack environment
- How do you test the program?

Ex3-Calculate STD

```
import math
Import numpy as np
data = [ 1, 2, 3, 4, 3, 2, 3, 4]
def mean ( input ):
        return sum(input)/float(len(input))
def sqr ( x ):
        return x*x
def std( input ):
        m = mean( input )
        var = [ sqr(i-m) for i in input ]
        return math.sqrt( sum(var)/float(len(var)) )
mean(data)
std(data)
assert std(data) == np.std(data)
```

- English to code
- Sequence of expression evaluation
- Visualize stack environment
- How do you test the program?

Questions?