Discussion 10

Scheme Data Abstractions

Aditya Balasubramanian
aditbala [at] berkeley [dot] edu

Slides available at `teaching.aditbala.com`

Announcements 📣

- Ants due Today
- HW05 due Today
- Scheme Project Today (7/28)
  - Checkpoint 1 due Tuesday (8/2)
  - Checkpoint 2 due Friday (8/5)
- Magic: the Lambda-ing due today!

Data Abstraction 

Data Abstraction (Scheme)

- What is Data Abstraction?
  - The idea of treating code as an object
  - User doesn't have to worry about how code is implemented
  - Classes in Python
- But what about Scheme?
  - Scheme does not have classes
  - Use data abstracations (functions)

Scheme Data Abstractions

- Constructors
  - functions that build the abstract data type
  - similar to `__init__` in Python
- Selectors
  - functions that retrieve information from the data type
  - Example is `car`, `cdr` in Scheme!

Cities

- Create an abstract data type for cities
- Keep track of `name`, `latitude`, `longitude`
- What information should our Constructor take in?
- `name`, `latitude`, `longitude`
- How many Selectors should we have?
- One for each piece of data

Cities (Implementation)

```
scm> (define berkeley (make-city 'Berkeley 122 37))
berkeley
scm> (get-name berkeley)
Berkeley
scm> (get-lat berkeley)
122
scm> (define new-york (make-city 'NYC 74 40))
new-york
scm> (get-lon new-york)
40
```

Worksheet

Tree Data Abstraction

- What do we need to keep track of?
- label and branches
- Syntax
  - Constructor
    - `(tree label branches)`
  - Selectors
    - `(label t)`
    - `(branches t)`

Tree ADT Implementation

```
scm> (define t (tree 5 (list (tree 4 nil) (tree 7 nil))))
t
scm> (label t)
5
scm> (label (car (branches t)))
4
scm> (label (car (cdr (branches t))))
7
```

Worksheet!

Thank you!!!

Attendance Form -> https://tinyurl.com/adit-disc10

Anon Feedback -> https://tinyurl.com/adit-anon