

CS1010S Tutorial 4

Sean Ng

AY2018/19 Sem 2, Week 6

Updated 2019-02-19 at 13:17:45

Materials: pengnam.github.io/1010s

Contact: seanngpengnam@u.nus.edu

Feedback From Coursemology

- DO NOT BREAK ABSTRACTION

- Recall how operations can be abstracted

Abstract Data Types

- Recall how operations can be abstracted

```
def get_gravitational_force(m1, m2, d):  
    G = 6.67408e-11  
    return (G * m1 * m2) / d**2
```

Abstract Data Types

- Recall how operations can be abstracted

```
def get_gravitational_force(m1, m2, d):  
    G = 6.67408e-11  
    return (G * m1 * m2) / d**2
```

- Similarly, we can abstract *data*

Abstract Data Types

```
def make_book(title, author):  
    def book(query):  
        if query == 0:  
            return title  
        else:  
            return author  
    return book
```

Abstract Data Types

```
def make_book(title, author):  
    def book(query):  
        if query == 0:  
            return title  
        else:  
            return author  
    return book
```

make_book: (str, str) \rightarrow Book

Abstract Data Types

```
def get_title(book):  
    return book(0)
```

```
def get_author(book):  
    return book(1)
```

Abstract Data Types

```
def get_title(book):  
    return book(0)
```

```
def get_author(book):  
    return book(1)
```

get_title: (Book) \rightarrow str

Abstract Data Types

```
def get_title(book):  
    return book(0)
```

```
def get_author(book):  
    return book(1)
```

get_title: (Book) \rightarrow str

get_author: (Book) \rightarrow str

Abstract Data Types

`make_book: (str, str) → Book`

`get_title: (Book) → str`

`get_author: (Book) → str`

Abstract Data Types

`make_book: (str, str) → Book`

`get_title: (Book) → str`

`get_author: (Book) → str`

```
def make_potter(series):  
    author = 'JKR'  
    if series == 1:  
        return make_book('Philosophers Stone', author)  
    elif series == 2:  
        return make_book('Chamber of S', author)  
    else:  
        return make_book('Prisoner of Azk', author)
```

Abstract Data Types

`make_book: (str, str) → Book`

`get_title: (Book) → str`

`get_author: (Book) → str`

```
def make_potter(series):  
    author = 'JKR'  
    if series == 1:  
        return make_book('Philosophers Stone', author)  
    elif series == 2:  
        return make_book('Chamber of S', author)  
    else:  
        return make_book('Prisoner of Azk', author)
```

`make_potter:`

Abstract Data Types

make_book: (str, str) \rightarrow Book

get_title: (Book) \rightarrow str

get_author: (Book) \rightarrow str

```
def make_potter(series):  
    author = 'JKR'  
    if series == 1:  
        return make_book('Philosophers Stone', author)  
    elif series == 2:  
        return make_book('Chamber of S', author)  
    else:  
        return make_book('Prisoner of Azk', author)
```

make_potter: (Number) \rightarrow

Abstract Data Types

make_book: (str, str) \rightarrow Book

get_title: (Book) \rightarrow str

get_author: (Book) \rightarrow str

```
def make_potter(series):  
    author = 'JKR'  
    if series == 1:  
        return make_book('Philosophers Stone', author)  
    elif series == 2:  
        return make_book('Chamber of S', author)  
    else:  
        return make_book('Prisoner of Azk', author)
```

make_potter: (Number) \rightarrow Book

Tutorial

- DO NOT BREAK ABSTRACTION