

S4: Higher Order Functions

CS1101S AY20/21 Sem 1

Studio 2D

Lee Wei Min

Table of Contents

- Info

Additional Information

- Missions / quests
- RA1 – past years
- Mastery checks
- Class part adjustment

Mastery Checks

- Random or pair up?
- Earlier the better
- Recorded
- Mastery check 1 – best within two weeks

Lambda Expressions

New kinds of expressions

$(\text{parameters}) \Rightarrow \text{expression}$

If there is only one parameter, you can write

$\text{parameter} \Rightarrow \text{expression}$

Scoping – Pre-declared functions

(1) Pre-declared names

The [Source §1](#) pages tell us what names are pre-declared, e.g. `math_floor`.

Scoping- Constants

(2) Constant declarations

The scope of a constant declaration is the closest surrounding pair of `{...}`, or the whole program, if there is none.

Scoping - Functions

(3) Parameters

The scope of the parameters of a lambda expression or function declaration is the body of the function.

```
function f(x, y, z) {  
    ... x ... y ... z ...  
}
```

```
(v, w, u) => ... v ... w ... u ...
```


Scoping- Functions

(4) Function name

The scope of the function name of a function declaration is as if the function was declared with `const`.

```
function f(x) {  
    ...  
}
```

as if we wrote

```
const f = ...;
```

Coin Change

Given: Different kinds of coins (unlimited supply)

Given: Amount of money in cents

Wanted: Number of ways to change amount into coins

Coin Change – First Denom?



```
1  function first_denomination(kinds_of_coins) {  
2      return kinds_of_coins === 1  
3          ? 5  
4          : kinds_of_coins === 2  
5          ? 10  
6          : kinds_of_coins === 3  
7          ? 20  
8          : kinds_of_coins === 4  
9          ? 50  
10         : kinds_of_coins === 5  
11         ? 100  
12         : 0;  
13 }
```

Coin Change- Skeleton



```
1  function cc(amount, kinds_of_coins) {  
2  
3  }
```

Coin Change – Base Cases



```
1  function cc(amount, kinds_of_coins) {  
2      return amount === 0  
3          ? 1  
4          : amount < 0 || kinds_of_coins === 0  
5          ? 0  
6      }
```

Coin Change – Subproblems



```
1 function cc(amount, kinds_of_coins) {  
2   return amount === 0  
3     ? 1  
4     : amount < 0 || kinds_of_coins === 0  
5     ? 0  
6     : cc(amount - first_denomination(kinds_of_coins), kinds_of_coins) +  
7       cc(amount, kinds_of_coins - 1);  
8 }
```

Studio Sheet