

Higher-order Functions Review

Announcements

- HACK110 is Friday at 7pm!
 - Are you taking 401 this Spring? Do you want more CS? You should come!
 - Coming? Great! RSVP: <http://hack-110.com/>
- Review Material
 - Slide Decks
 - Topics Pages
 - Current Worksheet (Key released tomorrow)
 - UTA Videos

1. Which functions are actually Funcy?

```
interface Funcy<A, B> {  
    (a: B, b: A): B;  
}  
  
let a: Funcy<number, string> = (s: string, t: number): string => {  
    return "?";  
};  
  
let b: Funcy<string, number> = (s: string, t: number): string => {  
    return "?";  
};  
  
let c: Funcy<string, number> = (s: number, t: string): number => {  
    return 1;  
};  
  
let d: Funcy<number, number> = (s: number, t: number): number => {  
    return 1;  
};
```

2. Of functions of b, c, d, and e, which are not exactly equivalent to function a?

```
let a: Transform<string, number> = (input: string): number => {  
  return input.length;  
};
```

```
let b: Transform<string, number> = (input) => {  
  return input.length;  
};
```

```
let c: Transform<string, number> = (input) => input.length;
```

```
let d = (input: string): number => {  
  return input.length;  
};
```

```
let e = (input: string): number => input.length;
```

3. What is the result of this call to `filter`?

```
filter(listify(1, 2, 3, 4), (item) => item % 2 === 1)
```

4. What is the result of this call to `map`?

```
map(listify(1, 3), (item) => item * 2)
```

5. Answer a couple questions related to the signature and an example call to reduce:

Signature:

```
let reduce = <T, U> (xs: Node<T>, f: Reducer<T, U>, memo: U): U => {
```

Call:

```
reduce(listify(2,4,6), (memo, item) => item + memo, "")
```

6. "Just for funcies" -- Given the code, draw an environment diagram at the breakpoint. Once drawn, answer the questions on PollEv.com

```
00 interface Funcy {
01     (a: number, b: number): number;
02 }
03
04 export let main = async () => {
05     let a = 16;
06     let b = 2;
07     let c = justF((or, funcies) => or - funcies, b, a);
08     // Breakpoint here
09     print(c);
10 };
11
12 let justF = (f: Funcy, a: number, b: number): number => {
13     return f(b / a, a);
14 };
15
16 main();
```


7. Answer the questions on PollEv.

```
let zip = <T> (a: Node<T>, b: Node<T>): Node<T[]> => {
  if (a === null || b === null) {
    return null;
  } else {
    let pair: T[] = [first(a), first(b)];
    return cons(pair, zip(rest(a), rest(b)));
  }
};
```

```
let ariana = listify("thank", "next");
let grande = listify("you", "!!!");
let fire = zip(ariana, grande);
```