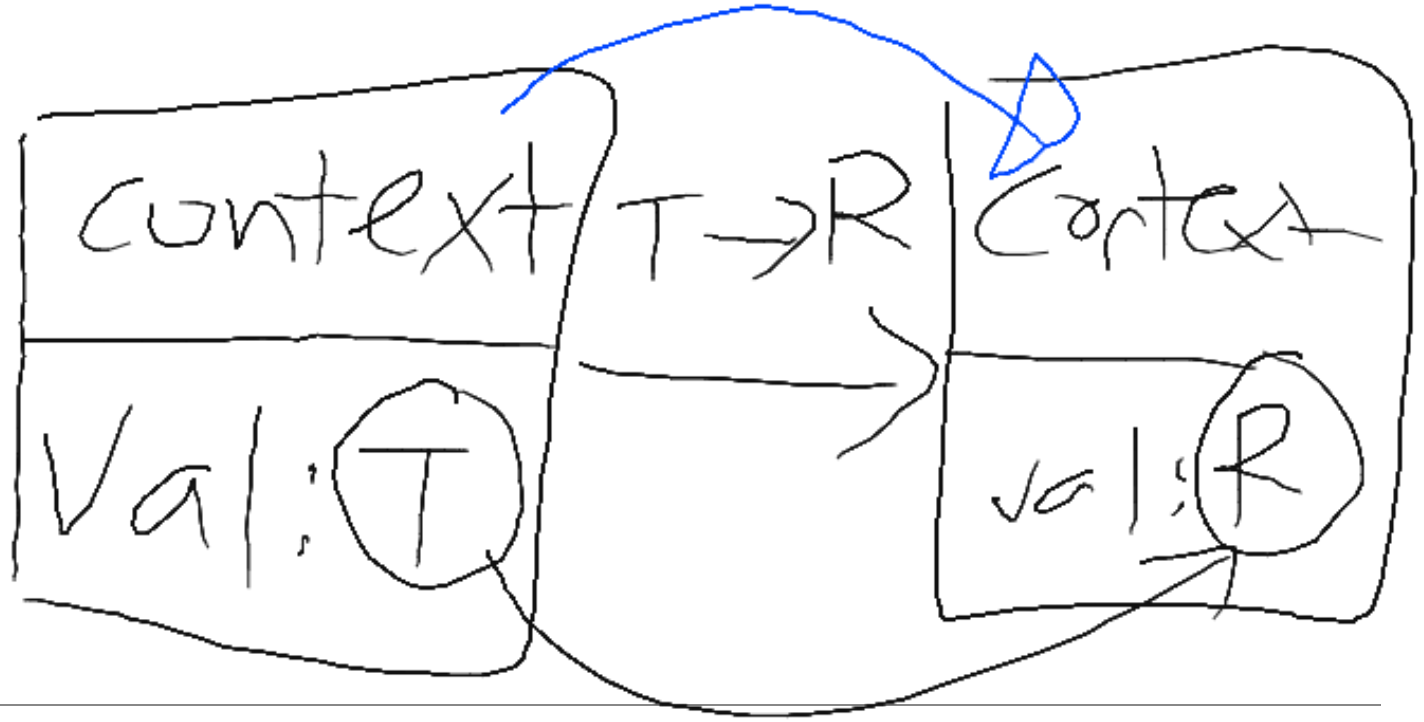
CS2030S

Programming Methodology II

Recitation 09

Q1



Monad & Functor

Q1

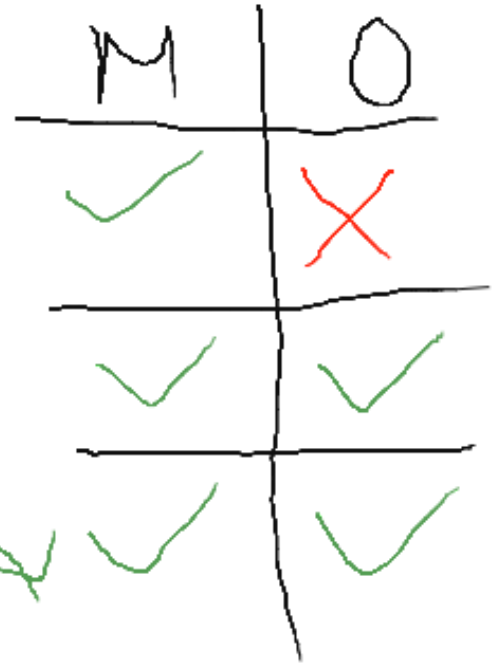
Recap

$(a+b)+c = a+(b+c) = a+b+c$

Monad

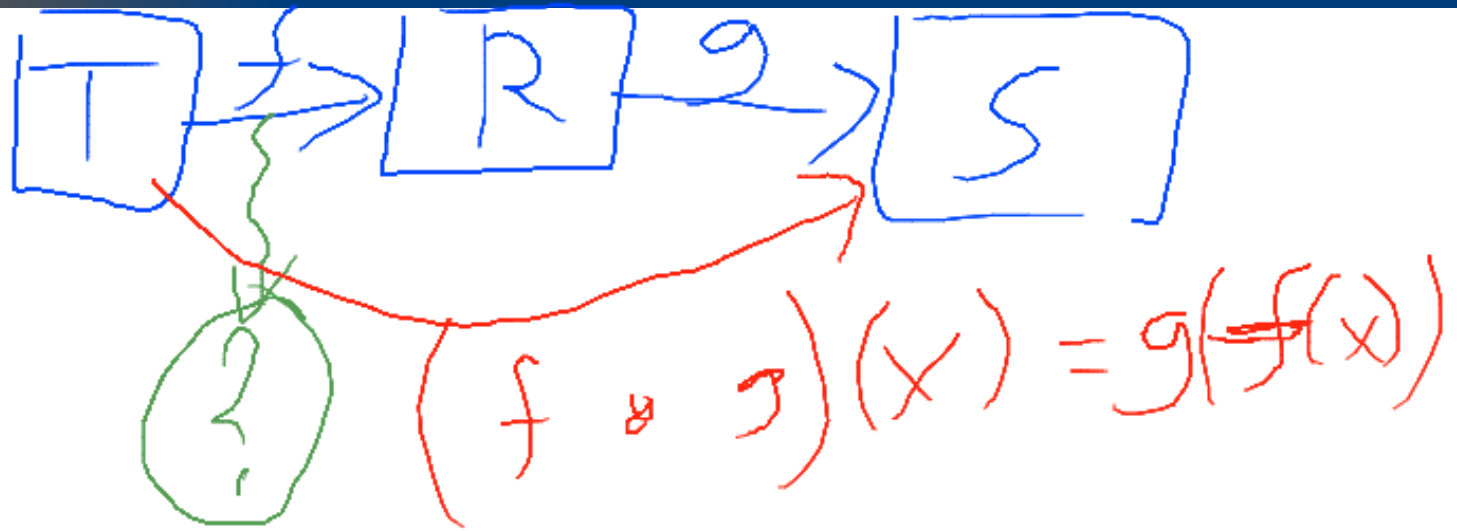
> **Definition**
>
> A **monad** has two methods `of` *(some, none, ok, err etc)* and `flatMap` *(or next, bind, etc)* and obeys **three** laws:

1. **Left Identity Law:**
   - `Monad.of( x ).flatMap(y -> f(y) )` ≡ `f(y)`
2. **Right Identity Law:**
   - `monad .flatMap(y -> Monad.of(y))` ≡ `monad`
3. **Left Identity Law:**
   - `monad.flatMap(x -> f(x) ) .flatMap(x -> g(x))`
     ≡
     `monad.flatMap(x -> f(x).flatMap(x -> g(x))) )`



A SSOC

Q1



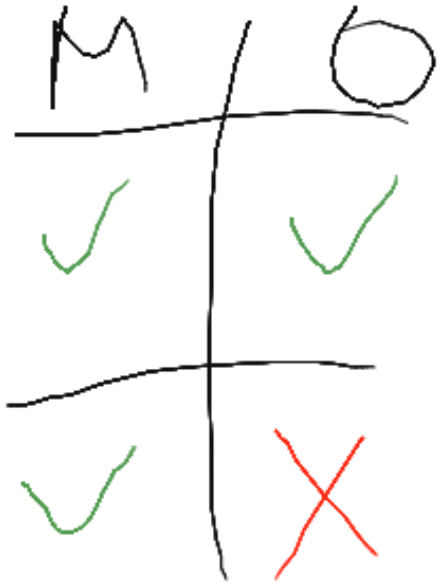
Recap

Functor

Definition

A **monad** has two methods `of` *(`some`, `none`, `ok`, `err` etc)* and `map` *(or `fmap`, `transform`, etc)* and obeys **two** laws:

1. **Identity Morphism:**
   - `functor.map(x -> x)` ≡ `functor`
2. **Composition Morphism:**
   - `functor.map(x -> f(x)).map(x -> g(x))`
     ≡
     `functor.map(x -> g(f(x)))`



Q1

Recap
NULL
- Maybe

this.val

Dealing with NULL Values

Maybe< T >

Method	NULL	Non-NULL
`static some(val)`	`NULL`	`Maybe`
`flatMap(f : T -> Maybe)`	`f.apply(...)`	
`map(f : T -> R)`	`Maybe.some(f.apply(...))`	

Q1

Recap
NULL
- *Maybe*
- *Optional*

Dealing with NULL Values

Optional< T >

Java SE 17 & JDK 17

Module java.base
Package java.util

Class Optional<T>

java.lang.Object
    java.util.Optional<T>

Type Parameters:

T - the type of value

```
public final class Optional<T>
extends Object
```

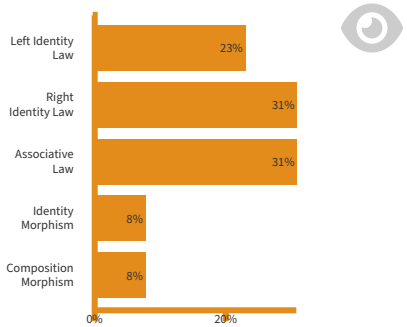
A container object which may or may not contain a non-`null` value. If a value is present, `isPresent()` returns `true`. If no value is present, the object is considered *empty* and `isPresent()` returns `false`.

Q1

Recap
NULL
Questions
- Maybe

Maybe

> **Question**
>
> Which of the three Monad laws and two Functor laws do `Maybe` obey?



Total Results: 13

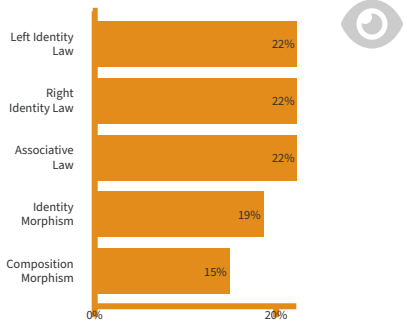
	Choice	Comment	
A	Left Identity Law		?
B	Right Identity Law		?
C	Associative Law		?
D	Identity Morphism		?
E	Composition Morphism		?

Q1

Recap
NULL
Questions
- *Maybe*
- *Optional*

Optional

> **Question**
>
> Which of the three Monad laws and two Functor laws do `Optional` obey?



Total Results: 27

	Choice	Comment	
A	Left Identity Law		?
B	Right Identity Law		?
C	Associative Law		?
D	Identity Morphism		?
E	Composition Morphism		?

Q2

Stream

Q2

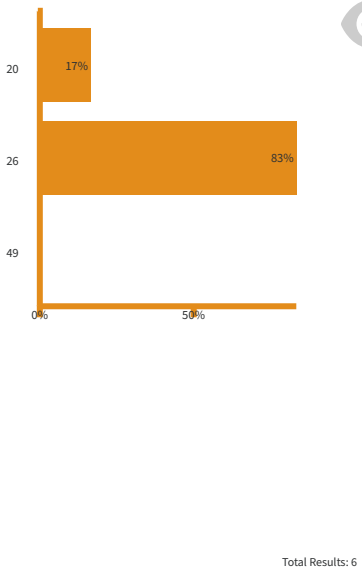
Stream

Stream

Question

What is the result of the following code?

```
Stream.of(1, 2, 3, 4)
  .reduce(0, (result, x) -> result * 2 + x);
```



	Choice	Comment	
A	20		?
B	26		?
C	49		?

Q3

Fibonacci

Q3

Fibonacci

Fibonacci

> **Definition**
>
> $F_1 = 1$
>
> $F_2 = 1$
>
> $F_n = F_{n-1} + F_{n-2}$

Recursive

```
int fibR(int n) {
  if (n <= 2) {
    return 1;
  } else {
    return fibR(n-1) + fibR(n-2);
  }
}
```

Iterative

```
int fibR(int n) {
  int f1 = 1; int f2 = 1;
  for (int i=0; i<n; i++) {
    int fn = f2 + f1; f2 = f1; f1 = fn;
  }
  return f2;
}
```

Q3

Fibonacci

Alternative Definition

Let F_1 to F_4 be known. Then

- $F_5 = F_3 + F_4$
- $F_6 = F_4 + F_5$
- $F_7 = F_5 + F_6$

Q3

Fibonacci
- Alternative

Fibonacci

Alternative Definition

Let F_1 to F_4 be known. Then

- $\boldsymbol{F_5} = F_3 + F_4$
- $\boldsymbol{F_6} = F_4 + \boldsymbol{F_5}$
- $F_7 = \boldsymbol{F_5} + \boldsymbol{F_6}$

Q3

Fibonacci
- Alternative

Fibonacci

Alternative Definition

Let F_1 to F_4 be known. Then

- $\mathbf{F_5} = F_3 + F_4$
- $\mathbf{F_6} = F_4 + \mathbf{F_3 + F_4}$
- $F_7 = \mathbf{F_3 + F_4} + \mathbf{F_4 + F_3 + F_4}$

Q3

Fibonacci
- Alternative

Fibonacci

Alternative Definition

Let F_1 to F_4 be known. Then

- $F_5 = F_3 + F_4$
- $F_6 = F_3 + F_4 + F_4$
- $F_7 = F_3 + F_3 + F_4 + F_4 + F_4$

Q3

Fibonacci
- Alternative

Fibonacci

Alternative Definition

Let F_1 to F_4 be known. Then

- $F_5 = \square F_3 + \square F_4$
- $F_6 = \square F_3 + \mathbf{2} \times F_4$
- $F_7 = \mathbf{2} \times F_3 + \mathbf{3} \times F_4$

Q3

Fibonacci

Alternative Definition

Let F_1 to F_4 be known. Then

- $F_5 = \mathbf{1} \times F_3 + \mathbf{1} \times F_4$
- $F_6 = \mathbf{1} \times F_3 + \mathbf{2} \times F_4$
- $F_7 = \mathbf{2} \times F_3 + \mathbf{3} \times F_4$

Q3

Fibonacci

Alternative Definition

Let F_1 to F_4 be known. Then

- $F_5 = \mathbf{F_1} \times F_3 + \mathbf{F_2} \times F_4$
- $F_6 = \mathbf{F_2} \times F_3 + \mathbf{F_3} \times F_4$
- $F_7 = \mathbf{F_3} \times F_3 + \mathbf{F_4} \times F_4$

Q3

Fibonacci
- Alternative

Fibonacci

Alternative Definition

Let F_1 to F_4 be known. Then

- $F_5 = \mathbf{F_1} \times F_3 + \mathbf{F_2} \times F_4$
- $F_6 = \mathbf{F_2} \times F_3 + \mathbf{F_3} \times F_4$
- $F_7 = \mathbf{F_3} \times F_3 + \mathbf{F_4} \times F_4$

Notes

- There are no more F_5 to F_7
- Given F_1 to F_4, we can generate each F_5 to F_7 independently

Q3

Fibonacci

Alternative Definition

Let F_1 to F_k be known. Then

- $F_{k+1} = \mathbf{F_1} \times F_{k-1} + \mathbf{F_2} \times F_k$
- $F_{k+2} = \mathbf{F_2} \times F_{k-1} + \mathbf{F_3} \times F_k$
- $F_{k+3} = \mathbf{F_3} \times F_{k-1} + \mathbf{F_4} \times F_k$

Q3

Fibonacci
- Alternative

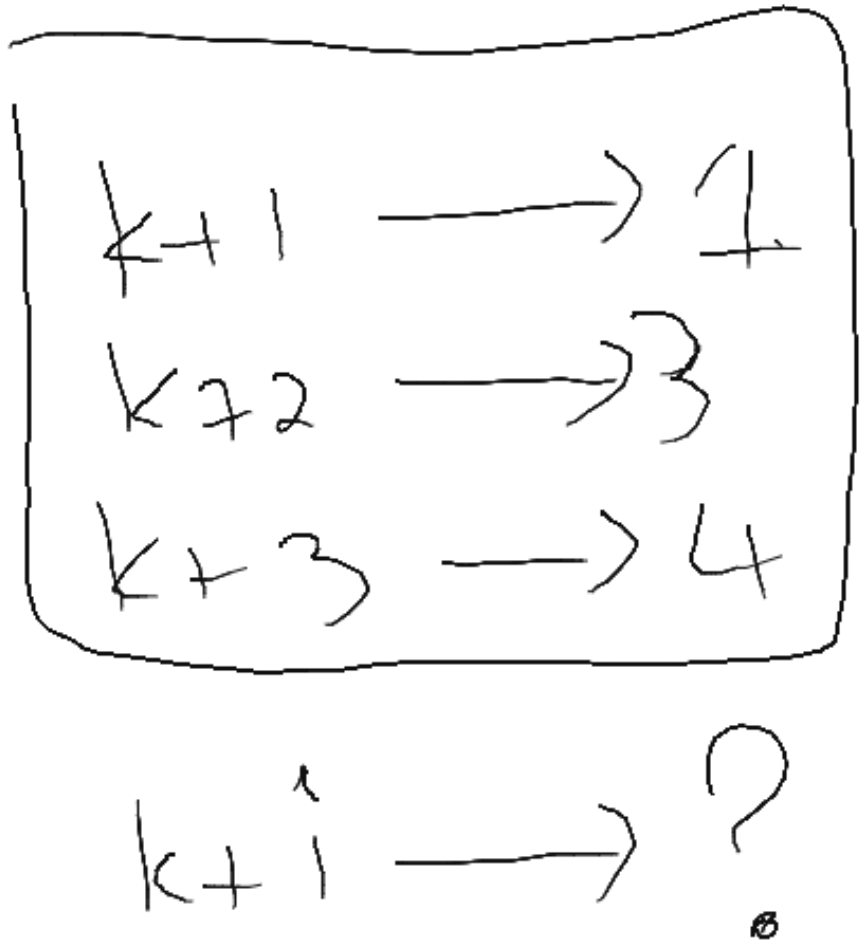
Fibonacci

Alternative Definition

Let F_1 to F_k be known. Then

- $F_{k+1} = \mathbf{F_1} \times F_{k-1} + \mathbf{F_2} \times F_k$
- $F_{k+2} = \mathbf{F_2} \times F_{k-1} + \mathbf{F_3} \times F_k$
- $F_{k+3} = \mathbf{F_3} \times F_{k-1} + \mathbf{F_4} \times F_k$

- :
- $F_{k+i} = \mathbf{F_i} \times F_{k-1} + \mathbf{F_{i+1}} \times F_k$



Q3

Fibonacci
- Alternative

Fibonacci

Alternative Definition

Let F_1 to F_k be known. Then

- $F_{k+1} = \mathbf{F_1} \times F_{k-1} + \mathbf{F_2} \times F_k$
- $F_{k+2} = \mathbf{F_2} \times F_{k-1} + \mathbf{F_3} \times F_k$
- $F_{k+3} = \mathbf{F_3} \times F_{k-1} + \mathbf{F_4} \times F_k$

- :
- $F_{k+i} = \mathbf{F_i} \times F_{k-1} + \mathbf{F_{i+1}} \times F_k$

- :
- $F_{2k-1} = \mathbf{F_{k-1}} \times F_{k-1} + \mathbf{F_k} \times F_k$

Q3

Fibonacci

Alternative Definition

Let F_1 to F_k be known. Then

- $F_{k+1} = \mathbf{F_1} \times F_{k-1} + \mathbf{F_2} \times F_k$
- $F_{k+2} = \mathbf{F_2} \times F_{k-1} + \mathbf{F_3} \times F_k$
- $F_{k+3} = \mathbf{F_3} \times F_{k-1} + \mathbf{F_4} \times F_k$

- :
- $F_{k+i} = \mathbf{F_i} \times F_{k-1} + \mathbf{F_{i+1}} \times F_k$

- :
- $F_{2k-1} = \mathbf{F_{k-1}} \times F_{k-1} + \mathbf{F_k} \times F_k$

Note

Given F_1 to F_k, we can compute F_{k+1} to F_{2k-1} in parallel

```
jshell> /exit
|    Goodbye
```