

CIS 5520

Advanced Programming

Fall 2024

- Work on IOExercise and MonoidFoldable today!



Today: Wed Sep 25

- HW #2 due Thursday (tomorrow)
- Monday: Read QuickCheck module & complete Gradescope quiz
- HW #3 (QuickCheck, purely functional data structures) released Monday, new partners
- Today: Two in-class exercises
 - IOExercise (discussion at 12:30)
 - MonoidFoldable (discussion at 1:10)

Semigroup/Monoid

```
class Semigroup a where                                -- associative binary operator
    (<>) :: a -> a -> a                                -- a <> (b <> c) == (a <> b) <> c
instance Semigroup [a] where
    (<>) = (++)
class Semigroup a => Monoid a where                  -- with identity element
    mempty :: a                                         -- a <> mempty == a
instance Monoid [a] where                             -- mempty <> a == a
    mempty = []
```

Foldable

class Foldable t **where**

foldMap :: Monoid m => (a -> m) -> t a -> m

...

instance Foldable [] **where**

foldMap :: Monoid m => (a -> m) -> [a] -> m

foldMap f = List.foldr (\x acc -> f x <> acc) mempty

toList :: Foldable t => t a -> [a] -- *this is an iterator in Haskell*