

CSC 447 - Concepts of Programming Languages

Map, Filter, and Fold Examples

Instructor: Eric J. Fredericks



Exercise: Process a List of Numbers

```
1 val xs = List(1, 2, 3, 4)
2 val ys = List(2, 3, 4, 5)
```

? Which snippets return `true` ?

1. `ys == xs.map(x => x+1)`
2. `ys == xs.map(_ + 1)`
3. `ys == xs.filter(x => x % 2 == 0)`
4. `ys == xs.fold(0)((x, y) => x+y)`

- Snippets 1 and 2



Exercise: Process a List of Numbers

```
1 val xs = List(1, 2, 3, 4)
2 val ys = List(2, 4)
```

? Which snippets return `true` ?

1. `ys == xs.map(x => x+1)`
2. `ys == xs.map(_ + 1)`
3. `ys == xs.filter(x => x % 2 == 0)`
4. `ys == xs.filter(_ % 2 == 0)`
5. `ys == xs.fold(0)((x,y) => x+y)`

- Snippets 3 and 4



Exercise: Process a List of Numbers

```
1 val xs = List(1, 2, 3, 4)
2 val ys = 10
```

? Which snippets return `true` ?

1. `ys == xs.map(x => x+1)`
2. `ys == xs.map(_ + 1)`
3. `ys == xs.filter(x => x % 2 == 0)`
4. `ys == xs.filter(_ % 2 == 0)`
5. `ys == xs.fold(0)((x,y) => x+y)`

- Snippet 5



Exercise: Process a List of Numbers

```
1 val xs = List(1, 2, 3, 4)
2 val ys = "01234"
```

? Which snippets return `true` ?

1. `ys == xs.map(x => x+1)`
2. `ys == xs.map(_ + 1)`
3. `ys == xs.filter(x => x % 2 == 0)`
4. `ys == xs.filter(_ % 2 == 0)`
5. `ys == xs.fold("0")((x, y) => x.toString+y)`

- Snippet 5



Exercise: Process a List of Numbers

```
1 val xs = List(1, 2, 3, 4)
2 val ys = "1, 2, 3, 4, "
```

? Which snippets return `true` ?

1. `ys == xs.map(x => x+1)`
2. `ys == xs.map(_ + 1)`
3. `ys == xs.filter(x => x % 2)`
4. `ys == xs.filter(_ % 2 == 0)`
5. `ys == xs.fold(",")((x,y) => x.toString+y)`
6. `ys == xs.map(x => x.toString + ",").fold("")(x,y) => x+y)`

- Snippet 6



Exercise: Process a List of Numbers

```
1 val xs = List(1, 2, 3, 4)
2 val ys = ",2,4,"
```

? Which snippets return `true` ?

1. `ys == xs.map(x => x+1)`
2. `ys == xs.map(_ + 1)`
3. `ys == xs.filter(x => x % 2)`
4. `ys == xs.filter(_ % 2 == 0)`
5. `ys == xs.fold(",")((x,y) => x.toString+y)`
6. `ys == xs.filter(_ % 2 == 0).map(x => x.toString + ",").fold(",")((x,y) => x+y)`
7. `ys == xs.filter(_ % 2 == 0).map(_.toString + ",").fold(",")(_ + _)`

- Snippets 6 and 7