Method Reference Cheat Sheet

object::instanceMethod. Given object ob with method meth:

```
x -> ob.meth(x)
```

can be written as

```
ob::meth
```

Example

Rewrite

```
        button.setOnAction(evt -> p.print(evt));
```

as

```
        button.setOnAction(p::print);
```

Class::staticMethod. Given a class `ClassName` and one of its static methods `meth()`

```
x -> ClassName.meth(x)
```

can be rewritten as

```
ClassName::meth
```

Example

Rewrite

```
    BiFunction<Integer, Integer, Double> f = (x,y) -> Math.pow(x, y);
```

as

```
    BiFunction<Integer, Integer, Double> f = Math::pow;
```

Class::instanceMethod. Given a class `ClassName` and one of its instance methods `meth()`

```
(x,y) -> x.meth(y)
```

can be rewritten as

```
ClassName::meth
```

Example

```
    (str1, str2) -> str1.compareToIgnoreCase(str2)
```

can be written as

```
    String::compareToIgnoreCase
```

Class::new. Given a class name `ClassName` and one of its constructors that accepts an argument of type T, `ClassName::new` is equivalent to `(T x) -> new ClassName(x)`. The type T and number of arguments are determined by context.

Example Given a List of Labels, the following produces a list of Button objects, with those labels

```
    Stream<Button> stream = labels.stream().map(Button::new);
```

Example Constructor reference `int[]::new` is equivalent to `len -> new int[len]`