

◁ Comparator < AnyType >

↳ generic interface provided by Java

- allows classes to have flexibility in "natural ordering"

• You (aka the Comparator) define what the "natural ordering" is for that Collection

- Note: we can use Comparators with PriorityQueues!

◁ Comparator EXAMPLE

Let's set up a Comparator that orders integers from largest to smallest, instead of the opposite

```
public class LargeComparator implements Comparator<Integer>
{
    public int compare (Integer lhs, Integer rhs)
    {
        if (lhs > rhs)
            return -1;
        if (lhs < rhs)
            return 1;
        return 0;
    }
}
```

Now we have to use it for our PriorityQueue!

```
PriorityQueue<Integer> p = new PriorityQueue<Int.>(11, new LargeComparator());
p.add(14);
p.add(2);
p.add(33);
for (Integer i: p)
    S.O.P(i); // OUTPUT: 33, 14 2
```