

```
1 #include "SortableIntVector.h"
2 #include "IntVector.h"
3 #include <cstdlib>
4
5 SortableIntVector::SortableIntVector(const int aArrayOfIntegers[],
6                                     size_t aNumberOfElements)
7     : IntVector(aArrayOfIntegers, aNumberOfElements) {}
8
9 void SortableIntVector::sort(Comparable aOrderFunction) {
10     size_t n = this->size();
11     for (size_t i = 0; i <= n - 1; i++) {
12         for (size_t j = n - 1; j >= i + 1; j--) {
13
14             // turns out you can just do this. If you flip it, it will      ↗
15             // sort in reverse and the reason is that we are following the sort ↗
16             // lambda in Main. so in essence, you could define aOrderFunction as ↗
17             // the lambda to check a > b, but that would go against the ↗
18             // definition of our Comparable here, so I won't do that.
19
20             if (aOrderFunction(this->get(j), this->get(j - 1))) {
21                 // swap(a[j], a[j - 1])
22                 this->swap(j, j - 1);
23             }
24         }
25     }
26 }
```