

Chapter 4: Generics

4.23

```
9      public static <GenericType extends Comparable<? super GenericType>>
10      GenericType min(GenericType a, GenericType b) {
11          int val = a.compareTo(b);
12
13          if (val < 0)
14          {
15              return a;
16          }
17
18          return b;
19      }
20
21      public static <GenericType extends Comparable<? super GenericType>>
22      GenericType max(GenericType a, GenericType b) {
23          int val = a.compareTo(b);
24
25          if (val < 0)
26          {
27              return b;
28          }
29          return a;
30      }
```

4.24

```
32      public static <GenericType extends Comparable<? super GenericType>>
33      GenericType min(GenericType [] a) {
34          GenericType minimum = a[0];
35
36          for (GenericType value : a)
37              if (min(minimum, value) == value)
38                  minimum = value;
39
40          return minimum;
41      }
```

4.29

```
1  public class GenericMemoryCell<AnyType extends Comparable<? super AnyType>>
2  implements Comparable<GenericMemoryCell<AnyType>>
3  {
4
5      private AnyType storedValue;
6
7      public AnyType read( ) { return storedValue; }
8      public void write( AnyType x ) { storedValue = x; }
9
10
11      @Override
12      public int compareTo(GenericMemoryCell<AnyType> o)
13      {
14          return this.storedValue.compareTo(o.storedValue);
15      }
16  }
```

Chapter 4: Generics

4.36

```
44     @Override
45     public int compareTo(Person o)
46     {
47         return this.name.compareToIgnoreCase(o.name);
48     }
```

Testing Output

```
C:\Users\Initec\Documents\Personal\cs2420_summer2023\Chapter4\Generics - VS Code Console
Starting tests
Testing 4.23/4.24
Finishing testing 4.23/4.24
Starting tests for 4.29 (Comparable<MemoryCell>)
Ending tests for 4.29
Starting tests for 4.36 (Comparable method in the Person class)
Finishing tests for 4.36
Ending tests
Press any key to continue . . .
```

findMax Method

- I kept this method the same while testing the compareTo method, but was getting warnings about using Comparable in a not type safe way. This was my solution to clearing those up:

```
52     public static Comparable<Person> findMax( Comparable<Person> [ ] a )
53     {
54         int maxIndex = 0;
55
56         for( int i = 1; i < a.length; i++ )
57             if( a[ i ].compareTo( (Person)a[ maxIndex ] ) > 0 )
58                 maxIndex = i;
59
60         return a[ maxIndex ];
61     }
```