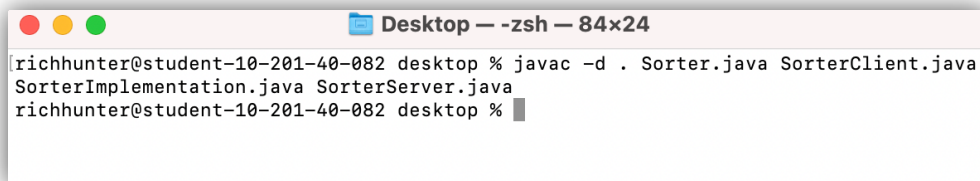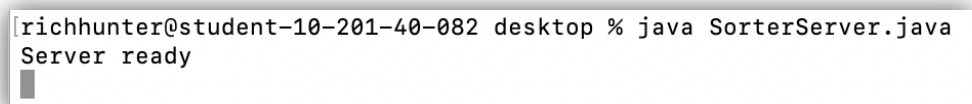1. Compile all java files using command: javac -d . Sorter.java SorterClient.java
   SorterImplementation.java SorterServer.java
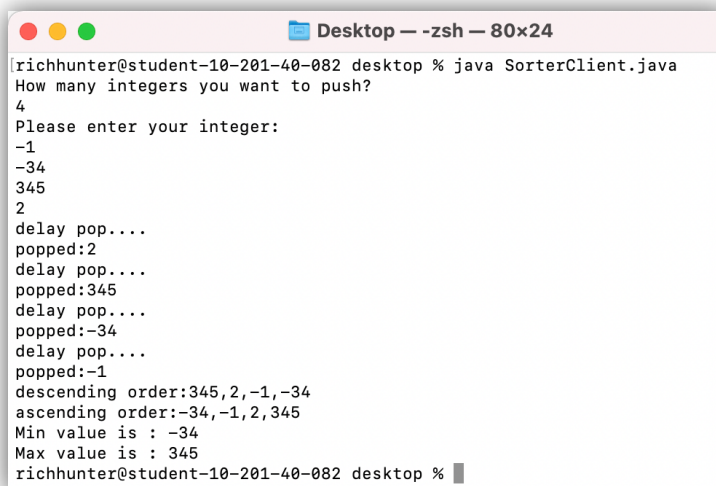
```
● ● ●                    📁 Desktop — -zsh — 84×24
[richhunter@student-10-201-40-082 desktop % javac -d . Sorter.java SorterClient.java
SorterImplementation.java SorterServer.java
richhunter@student-10-201-40-082 desktop % █
```

2. Start server using command: java SorterServer.java
   It shows "Server ready".

```
[richhunter@student-10-201-40-082 desktop % java SorterServer.java
 Server ready
 █
```

3. Open another terminal window to start client and push integer to stack

```
● ● ●                    📁 Desktop — -zsh — 80×24
[richhunter@student-10-201-40-082 desktop % java SorterClient.java
How many integers you want to push?
4
Please enter your integer:
-1
-34
345
2
delay pop....
popped:2
delay pop....
popped:345
delay pop....
popped:-34
delay pop....
popped:-1
descending order:345,2,-1,-34
ascending order:-34,-1,2,345
Min value is : -34
Max value is : 345
richhunter@student-10-201-40-082 desktop % █
```

The delay message will show on Server terminal window.

```
● ● ●              📁 Desktop — java SorterServer.java — 84×24
[richhunter@student-10-201-40-082 desktop % javac -d . Sorter.java SorterClient.java ]
SorterImplementation.java SorterServer.java
[richhunter@student-10-201-40-082 desktop % java SorterServer.java              ]
 Server ready
 waiting 10 milliseconds
 waiting 10 milliseconds
 waiting 10 milliseconds
 waiting 10 milliseconds
 █
```

Junit testing

You can test each method by clicking the run button

```java
    @Test
    void isEmpty() throws RemoteException {
        Sorter test = new SorterImplementation();

        //pushed integer 4 to stack, then stack is not empty and will return false
        test.pushValue(4);
        assertFalse(test.isEmpty());

        // this one for testing not push any value, and stack is empty and will return true
        // assertTrue(test.isEmpty());
```

Here is the result. If it has an error, then it will show up error with red line.

```
Run:     JunitTest.isEmpty ×

                                          ✔ Tests passed: 1 of 1 test – 155 ms
     ✔ Test Results              155 ms   /Library/Java/JavaVirtualMachines/jdk1.8.0_301.jdk/Conter
        ✔ JunitTest              155 ms
           ✔ isEmpty()           155 ms   Process finished with exit code 0
```

You can also test the whole page by clicking the run class button.

```
Run:     JunitTest ×

                                          ✔ Tests passed: 5 of 5 tests – 174 ms
     ✔ Test Results              174 ms   /Library/Java/JavaVirtualMachines/
        ✔ JunitTest              174 ms   6
           ✔ pushValue()         147 ms
           ✔ pop()                6 ms    waiting 2 milliseconds
           ✔ delayPop()          13 ms    2
           ✔ pushOperator()       6 ms    waiting 2 milliseconds
           ✔ isEmpty()            2 ms    4

                                          waiting 2 milliseconds
```

For method pushOperator () since it cannot return anything from void, I just write down the correct answer and check the difference with output.

```java
    @Test
    //cannot find any method to return void, so i just print out and check the difference
    void pushOperator() throws RemoteException {
        Sorter haha =  new SorterImplementation();
        String val1 = "1,2,3";
        System.out.println("Test 1 Expected:");
        System.out.println("descending order:3,2,1\n" +
                "ascending order:1,2,3\n" +
                "Min value is : 1\n" +
                "Max value is : 3");
        System.out.println("\n");
        System.out.println("Test 1 Actual:");
        haha.pushOperator(val1);
        System.out.println("-------------------");
```