

**Project Report:**  
**Critical Thinking 6 – Sort Students**

Alejandro Ricciardi

Colorado State University Global

CSC372: Programming 2

Professor: Dr. Vanessa Cooper

July 21, 2024

## Project Report:

### Critical Thinking 6 – Sort Students

This documentation is part of the Critical Thinking 6 Assignment from CSC372: Programming 2 at Colorado State University Global. This Project Report is an overview of the program's functionality and testing scenarios including console output screenshots. The program is coded in Java JDK-21; and is named Critical Thinking 6 (Sort Students). The program is composed of the Student class, SortingUtil class, and SortStudents which extends the Application class from the JavaFX library.

#### The Assignment Direction:

##### Option #1: Storing an ArrayList

Create a Java program that will store 10 student objects in an ArrayList, ArrayList<Student>. A student object consists of the following fields:

- int rollno
- String name
- String address

Implement two comparator classes to sort student objects by name and by rollno (roll number). Implement your own selection sort method and place your code in a separate Java source file. Do not use a sort method from the Java collections library.

**Students must use appropriate version control for all programmatic assignments created. GIT repositories should be established and screen captures of repositories submitted with each assignment.**

#### ⚠ My notes:

- I got permission from Dr. Cooper to use the JavaFX library to display the program outputs.
- I added my own icon to the window frame – logo.png
- **For the source code please see Student.java, SortingUtil.java, and SortStudents.java files.**

#### Program Description:

The Sort Students program sorts a list of students, allowing users to view and sort students by first name or roll number.

The program uses selection sort to sort the students.

## Git Repository

This is a picture of my GitHub page:

I use [GitHub](#) as my Distributed Version Control System (DVCS), the following is a link to my GitHub, [Omegapy](#).

My GitHub repository that is used to store this assignment is named [My-Academics-Portfolio](#) and the link to this specific assignment is: <https://github.com/Omegapy/My-Academics-Portfolio/tree/main/Programming-2-CSC372/Critical-Thinking-6>

## Classes Description:

- **Student Class:**

The class creates a student object with rollno (roll number), name, and address.

- **SortingUtil Class**

The class is a utility class providing sorting algorithms for Student objects.

In this version, the class only provides the selection sort algorithm.

The class also contains the inner classes:

- o The NameComparator class implements the Comparator Interface to compare student names.
- o The RoolNoComparator class implements the Comparator Interface to compare student roll numbers.

- **SortStudents Class:**

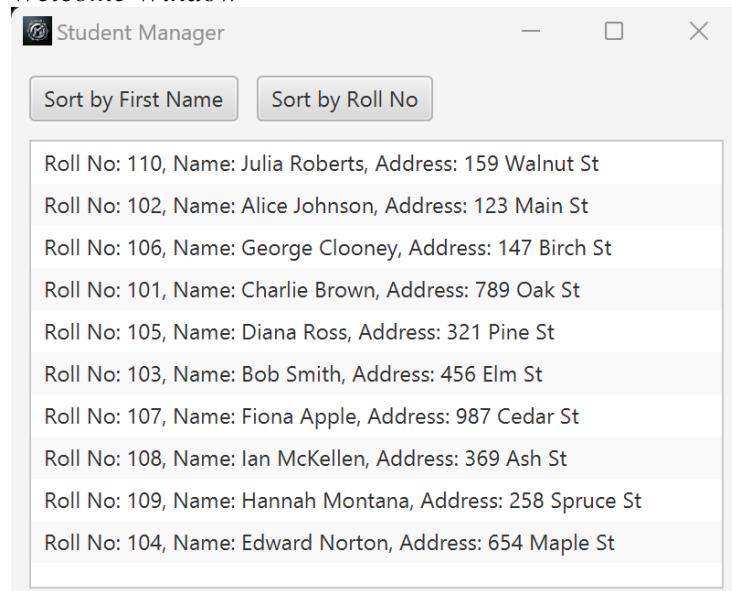
The class sorts Students and displays results, extends JavaFX's Application class, and contains the main method.

## Screenshot

### Program Functionality

**Figure 1**

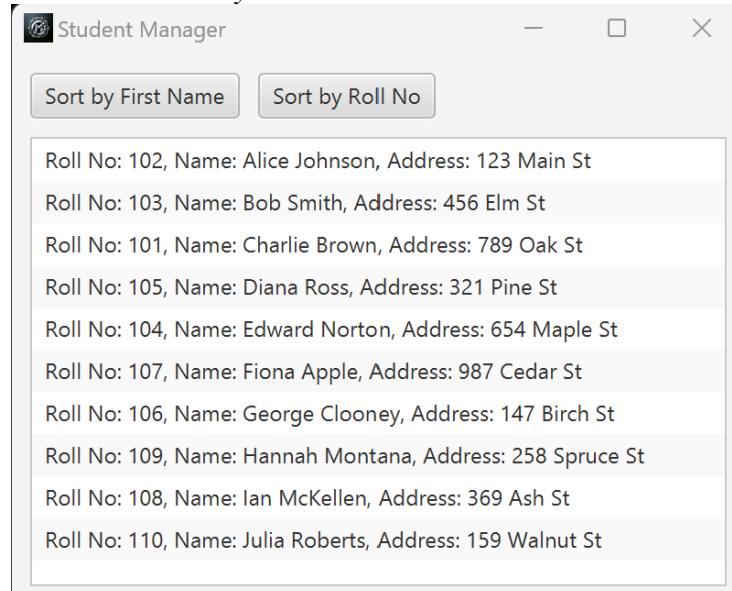
*Welcome Window*



*Note:* The students are not sorted.

**Figure 2**

*Students Sorted by First Names*



*Note:* The students are sorted by first names (names).

*Continue next page*

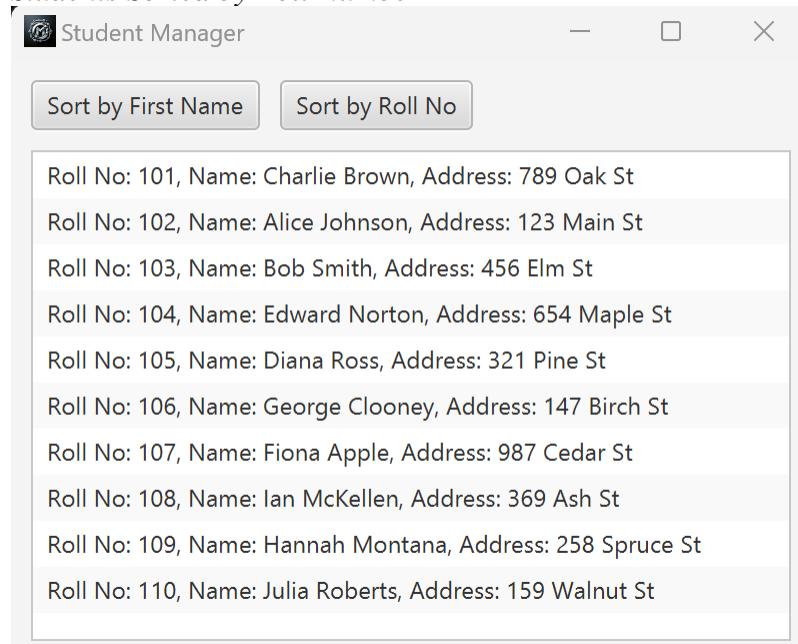
**Figure 3***Console Students Sorted by First Names Selection Sort Steps*

```
Sorted by First Name
-----
minIndex (i): 0
-----
i: 0, j: 1
minIndex (j): 1
-----
Swapping Index (i): 0 with index (j): 1
Swapping Object (i): Roll No: 110, Name: Julia Roberts, Address: 159 Walnut St
with Object (j): Roll No: 102, Name: Alice Johnson, Address: 123 Main St
-----
minIndex (i): 1
-----
i: 1, j: 2
minIndex (j): 2
-----
i: 1, j: 3
minIndex (j): 3
-----
i: 1, j: 5
minIndex (j): 5
-----
Swapping Index (i): 1 with index (j): 5
Swapping Object (i): Roll No: 110, Name: Julia Roberts, Address: 159 Walnut St
with Object (j): Roll No: 103, Name: Bob Smith, Address: 456 Elm St
-----
minIndex (i): 2
-----
i: 2, j: 3
minIndex (j): 3
-----
Swapping Index (i): 2 with index (j): 3
Swapping Object (i): Roll No: 106, Name: George Clooney, Address: 147 Birch St
with Object (j): Roll No: 101, Name: Charlie Brown, Address: 789 Oak St
-----
minIndex (i): 3
-----
i: 3, j: 4
minIndex (j): 4
-----
Swapping Index (i): 3 with index (j): 4
Swapping Object (i): Roll No: 106, Name: George Clooney, Address: 147 Birch St
with Object (j): Roll No: 105, Name: Diana Ross, Address: 321 Pine St
-----
minIndex (i): 4
-----
i: 4, j: 6
minIndex (j): 6
-----
```

```
i: 4, j: 9
minIndex (j): 9
-----
Swapping Index (i): 4 with index (j): 9
Swapping Object (i): Roll No: 106, Name: George Clooney, Address: 147 Birch St
with Object (j): Roll No: 104, Name: Edward Norton, Address: 654 Maple St
-----
minIndex (i): 5
-----
i: 5, j: 6
minIndex (j): 6
-----
Swapping Index (i): 5 with index (j): 6
Swapping Object (i): Roll No: 110, Name: Julia Roberts, Address: 159 Walnut St
with Object (j): Roll No: 107, Name: Fiona Apple, Address: 987 Cedar St
-----
minIndex (i): 6
-----
i: 6, j: 7
minIndex (j): 7
-----
i: 6, j: 8
minIndex (j): 8
-----
i: 6, j: 9
minIndex (j): 9
-----
Swapping Index (i): 6 with index (j): 9
Swapping Object (i): Roll No: 110, Name: Julia Roberts, Address: 159 Walnut St
with Object (j): Roll No: 106, Name: George Clooney, Address: 147 Birch St
-----
minIndex (i): 7
-----
i: 7, j: 8
minIndex (j): 8
-----
Swapping Index (i): 7 with index (j): 8
Swapping Object (i): Roll No: 108, Name: Ian McKellen, Address: 369 Ash St
with Object (j): Roll No: 109, Name: Hannah Montana, Address: 258 Spruce St
-----
minIndex (i): 8
```

*Continue next page*

**Figure 4**  
*Students Sorted by Roll Number*



*Note:* The students are sorted by roll numbers.

**Figure 5**  
*Console Students Sorted by Roll Number Selection Sort Steps*

```

-----  

Sorted by Roll No  

-----  

-----  

minIndex (i): 0  

-----  

i: 0, j: 2  

minIndex (j): 2  

-----  

Swapping Index (i): 0 with index (j): 2  

Swapping Object (i): Roll No: 102, Name: Alice Johnson, Address: 123 Main St  

with Object (j): Roll No: 101, Name: Charlie Brown, Address: 789 Oak St  

-----  

minIndex (i): 1  

-----  

i: 1, j: 2  

minIndex (j): 2  

-----  

Swapping Index (i): 1 with index (j): 2  

Swapping Object (i): Roll No: 103, Name: Bob Smith, Address: 456 Elm St  

with Object (j): Roll No: 102, Name: Alice Johnson, Address: 123 Main St  

-----  

minIndex (i): 2  

-----  

minIndex (i): 3
-----
```

```
i: 3, j: 4
minIndex (j): 4
-----
Swapping Index (i): 3 with index (j): 4
Swapping Object (i): Roll No: 105, Name: Diana Ross, Address: 321 Pine St
with Object (j): Roll No: 104, Name: Edward Norton, Address: 654 Maple St
-----
minIndex (i): 4
-----
minIndex (i): 5
-----
i: 5, j: 6
minIndex (j): 6
-----
Swapping Index (i): 5 with index (j): 6
Swapping Object (i): Roll No: 107, Name: Fiona Apple, Address: 987 Cedar St
with Object (j): Roll No: 106, Name: George Clooney, Address: 147 Birch St
-----
minIndex (i): 6
-----
minIndex (i): 7
-----
i: 7, j: 8
minIndex (j): 8
-----
Swapping Index (i): 7 with index (j): 8
Swapping Object (i): Roll No: 109, Name: Hannah Montana, Address: 258 Spruce St
with Object (j): Roll No: 108, Name: Ian McKellen, Address: 369 Ash St
-----
minIndex (i): 8
```

As shown in Figure 1 through Figure 5 the program runs without any issues displaying the correct outputs as expected.