# Overview

In this lab, you will implement the four different sorts:

1. **Bubble Sort**
2. **Selection Sort (optional)**
3. I**nsertion Sort**
4. **Merge Sort (optional)**

# Objectives

* Practice extending abstract classes
* Implement search algorithms

Starter Code

* <http://cs.unh.edu/~cs416/public/20L>

# Set-up

1. **Extend the abstract class Sort** in two classes: 1) **BubbleSort**, and 2) **InsertionSort**.
   1. You will also have to create classes for **SelectionSort** and **MergeSort**, but will not be required to fully implement those algorithms. You can just create stubs for them, and can use the classes to practice those algorithms if time allows.
2. The task is to read in a text file of “sample data” and then sort it. The file format is an id of type integer and measurement of type double separated by whitespace on each line. The **Sample** class is already created for you and the **Sort** class handles loading the file through its constructor. It populates an ArrayList. A string containing the file name is passed to the constructor. YOU WILL NEED TO SUBMIT BOTH OF THESE FILES. DO NOT MODIFY THEM.
3. For each class you will need to include a constructor that takes type String. Hint: This doesn’t require much work on your part for the code, you can explicitly call the parent constructor to do the work. You can use the keyword **super** to call the parent class' constructor.
4. The only **public method** you need to implement is **sort** for each class. You can implement additional methods as “helpers” but this is not required. **You may not use any built-in java methods for sorting**, you must implement each of the sorts with their corresponding class.

Continued on next page

1. Read the comments in the Sort class for any specific requirements on implementing the sort.
2. You may use the dat.csv file to look at a sample format for input and for testing while developing.

Submit to Gradescope :

* BubbleSort.java
* InsertionSort.java
* SelectionSort.java
* MergeSort.java