Java

Here is a sample Java implementation that sorts an ArrayList of numbers.

```
import java.util.ArrayList;
import java.util.Random;

public class QuickSort {

    public static final int NUMBERS_TO_SORT = 25;

    public QuickSort() {
    }

    public static void main(String[] args) {
        ArrayList<Integer> numbers = new ArrayList<Integer>();
        Random rand = new Random();
        for (int i = 0; i < NUMBERS_TO_SORT; i++)
            numbers.add(rand.nextInt(NUMBERS_TO_SORT + 1));
        for (int number : numbers)
            System.out.print(number + " ");
        System.out.println("\nBefore quick sort\n\n");
        for (int number : quicksort(numbers))
            System.out.print(number + " ");
        System.out.println("\nAfter quick sort\n\n");
    }

    public static ArrayList<Integer> quicksort(ArrayList<Integer> numbers) {
        if (numbers.size() <= 1)
            return numbers;
        int pivot = numbers.size() / 2;
        ArrayList<Integer> lesser = new ArrayList<Integer>();
        ArrayList<Integer> greater = new ArrayList<Integer>();
        int sameAsPivot = 0;
        for (int number : numbers) {
            if (number > numbers.get(pivot))
                greater.add(number);
            else if (number < numbers.get(pivot))
                lesser.add(number);
            else
                sameAsPivot++;
        }
        lesser = quicksort(lesser);
        for (int i = 0; i < sameAsPivot; i++)
            lesser.add(numbers.get(pivot));
```

```
        greater = quicksort(greater);
        ArrayList<Integer> sorted = new ArrayList<Integer>();
        for (int number : lesser)
            sorted.add(number);
        for (int number: greater)
            sorted.add(number);
        return sorted;
    }
}
```

Python

Using list comprehensions:

```
def qsort(L):
  if L == []: return []
  return qsort([x for x in L[1:] if x< L[0]]) + L[0:1] + \
         qsort([x for x in L[1:] if x>=L[0]])
```