Michael Xavier Canonizado

BSCS 2A – Object Oriented Programming

```java
public class CmdSorting {
        public static void main(String args[]) {
            int itemsCount = args.length;
            int unsortedItems[] = new int[itemsCount];

            // Load numbers from args to array
            for (int i = 0; i < itemsCount; i++) {
                unsortedItems[i] = Integer.parseInt(args[i]);
            }

            // Bubble Sort
            for (int i = 0; i < unsortedItems.length; i++) {
               for (int j = 0; j < itemsCount-1; j++) {
                    if (unsortedItems[j] > unsortedItems[j+1]) {
                        int temp = unsortedItems[j];
                        unsortedItems[j] = unsortedItems[j+1];
                        unsortedItems[j+1] = temp;
                    }
                }
            }

            // Print result
            System.out.print("\n\nSorted numbers:\n");
            for (int i = 0; i < unsortedItems.length; i++) {
                System.out.print(unsortedItems[i]+",");
            }
        }
}
```

```
Student@DESKTOP-3SVV1VQ MINGW64 /c/Student Root/BSCS2A-C
ANONIZADO/Lab 2
$ javac CmdSorting.java

Student@DESKTOP-3SVV1VQ MINGW64 /c/Student Root/BSCS2A-C
ANONIZADO/Lab 2
$ java CmdSorting 34 7 82 15 63 91 24 48 76 3 59 12 90 4
4 28 67 53 8 71 19 85 36 1 99 52 22 39 86 33 17 40 72 54
 95 2 38 66 25 11 80 45 77 5 60 32 13 30 99 74 46 29 10


Sorted numbers:
1,2,3,5,7,8,10,11,12,13,15,17,19,22,24,25,28,29,30,32,33
,34,36,38,39,40,44,45,46,48,52,53,54,59,60,63,66,67,71,7
2,74,76,77,80,82,85,86,90,91,95,99,99,
```