

## 7143CEM Programming for Data Science — Portfolio (draft version)

---

### Task 1. Explain, critique, comment and debug code (ILO2)

Suppose you have found the following (incomplete) Python code on a scrap of paper. It isn't quite finished and contains some errors.

```
def doit(L)
    s = True
    while (s)
        s = False
        for i in range(1,len(L)-3)
            if (L[i] <= L[i+1])
                # now swap values L[i] and L[i+1]
                s = True
        for i in range(len(L)-3,1,-1)
            if (L[i] <= L[i+1])
                # now swap values L[i] and L[i+1]
                s = True
```

- (1) Give a detailed critique of the incomplete code given above. Implement (in Python) the two lines that swap values, fix any syntax errors so that the Python code runs, ensure the code follows good Python coding style, improve all the variable names, improve the function name, and add useful comments. *Please annotate the existing code to show the syntax errors and provide a corrected version of the code. Do not fix any semantic errors and be careful about the indenting already given.*

[10 marks]

- (2) Add appropriate print statements to the Python code from part (1) to generate (directly from the code) a clear nontrivial example illustrating the algorithm running. The level of detail should show what comparisons are being made and what decisions are being taken. *Please provide both code (with print statements added) and the output from the code running in a way that illustrates the algorithm in detail but also gives insights into how it works. You can also add additional text explanation to your example. You may find it useful to use some indenting of the output using spaces in your print statements.*

[5 marks]

- (3) Insert additional Python code to count the number of comparisons and swaps of elements in the list. Make improvements (with justification) to the code to improve its effectiveness (so that it actually does solve the task correctly, now fixing any semantic errors) and efficiency (solves the task using fewer operations). Also, explain what task the code is attempting to solve (one sentence) and the higher-level idea behind the algorithm (one sentence). Explain how the resulting algorithm is similar to (and different from) one of Selection, Insertion or Bubble sort. *Clearly state the criteria you are using to judge your improvements. Test your final Python code on some more complex examples.*

[10 marks]