# CSC1108: Data Structures and Algorithms

## Tutorial 1: Python Programming

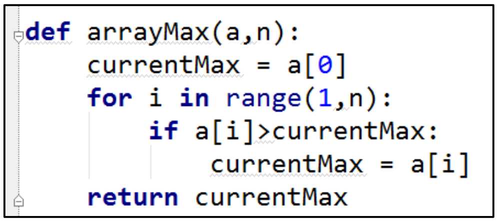
Q1. Write an algorithm that returns the index of the first occurrence of the largest element in an array s[0],…,s[n-1].

Q2. Write an algorithm that reverses the array s[0],…,s[n-1].

Q3. Write an algorithm that output the smallest and the second smallest values in the array s[0],…s[n-1]. Assume that n>1 and the values in the array are distinct.

Q4. Given an array s[0],…,s[n-1] such that n > 1 and s[i] ≤ s[i+1] for all i. Write an algorithm that inserts an input value x into the array so that s[i] ≤ s[i+1] for all i.

Q5. An algorithm for finding the maximum element of an array is in the following



```
def arrayMax(a,n):
    currentMax = a[0]
    for i in range(1,n):
        if a[i]>currentMax:
            currentMax = a[i]
    return currentMax
```

Determine the number of times that the statement “currentMax = a[i]” will be executed in the best case and in the worst case.

Q6. Implement a class called Student in Python which has the following attributes and operations:

Attributes:

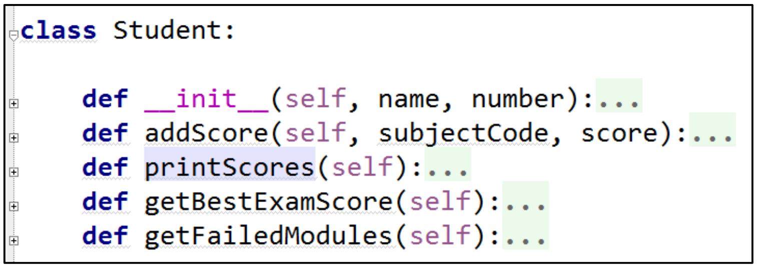
- Name
- Student number
- A list of pairs (subjectCode: examScore), for example: (“ICT1008”:96, “ICT1002”:100), of all the exam scores that the student has obtained so far.

Operations:

- getBestExamScore(): returns the subject code for which the student obtained the best exam score.
- getFailedModules(): print a list of subject codes for which the student has failed the exams (i.e. exam score < 40).

- addScore(subjectCode, examScore): adding the pair (subjectCode, examScore) to the list.
- printScore(): print the student’s name together with all the pairs (subjectCode, examScore) in the list. For e.g.: Cristal {'ICT1008': 96, 'ICT1002': 78}

The below figure shows an API of the Student class:



```
class Student:

    def __init__(self, name, number):...
    def addScore(self, subjectCode, score):...
    def printScores(self):...
    def getBestExamScore(self):...
    def getFailedModules(self):...
```