

## Assignment No. 2.

Name - Sumit Gulab Bamare

Div - A Roll no - 8

Sub - Data Structure Lab.

Aim - To illustrate the various functions in python.

Problem statement - Write a python program to store marks scored in subject "Fundamental of Data Structure" by N students in the class. Write functions to ~~complete~~ compute following:

- The average score of class
- Highest score & lowest score of class
- Count of students who were absent for the test
- Display mark with highest frequency.

Learning Objectives:

To understand basic techniques & strategies of algorithm analysis, the memory requirement for various data structures using the concepts of python.

Learning Outcome:

Students will be able to use algorithms on various linear data structure using sequential organization to solve real life problems

Theory: Lists are one of the most powerful tools in python. They are just like the arrays declared in other languages. But the most powerful thing is that list need not be always homogenous. A single list can contain strings, integers, as well as objects. Lists ~~are~~ can also be used for implementing



stacks & queues. Lists are mutable, i.e. they can be altered once declared.

A tuple is a sequence of immutable Python objects. Tuples are just like lists with the exception that tuples cannot be changed once declared. Tuples are usually faster than lists.

Input: Enter marks scored in subject " Fundamental of Data Structure " by N students in the class.

Output: Average score, highest score, lowest score, count of absentees, marks with highest frequency.

Algorithm/Pseudocode:

- The average score of class.

```
def find_average_score_of_class(A):
```

```
    sum = 0
```

```
    for i in range(len(A)):
```

```
        if (A[i] != -1):
```

```
            sum = sum + A[i]
```

```
    avg = sum / len(A)
```

```
    display_marks(A)
```

```
    print("\nAverage score of class is %.2f\n\n", avg)
```

• Highest score & lowest score of class.

```
def find_highest_and_lowest_score_of_class(A):
```

```
    max = -1
```

```
    min = 31
```

```
    for i in range(1, len(A)):
```

```
        if (max < A[i]):
```

```
            max = A[i]
```

```
            max_ind = i
```

```
        if (min > A[i] and A[i] != -1):
```

```
            min = A[i]
```

```
            min_ind = i
```

```
    display_marks(A)
```

```
    print("Highest Marks score of class is %d scored by student  
%d" % (max, max_ind+1))
```



Count of students who were absent for the test:

```
def find_count_of_absent_students(A):
    count = 0
    for i in range(len(A)):
        if (A[i] == -1):
            count += 1
    display_marks(A)
    print("\t Absent Student Count = %d" % count)
```

Display mark with highest frequency:

```
def display_mark_with_highest_frequency(A):
    freq = 0
    for i in range(len(A)):
        count = 0
        if (A[i] != -1):
            for j in range(len(A)):
                if (A[i] == A[j]):
                    count += 1
            if (freq < count):
                Marks = A[i]
                freq = count
    display_marks(A)
    print("\n Marks with highest frequency is %d (%d)" % (Marks, freq))
```

Software required: Open source Python, Programming tool like Jupyter, Notebook, Pycharm, Spyder.

Conclusion: Thus, we have studied the implementation of various python operations.