

STUDENT NAME: **YASH CHAUHAN**

STUDENT ROLL NUMBER: **28**

SECTION: **CSE C-1**

STUDENT REG NUMBER: **230905194**

WEEK NO: 4

DATE: **28TH JAN 2025**

Lab No. 4 – **BOOTSTRAP PYTHON BASICS (HOMEWORK)**

(ALL CODES WERE WRITTEN IN IPYNB (JUPYTER NOTEBOOK))

Q1. Write a python program to select smallest element from a list in an expected linear time.

```
arr = [12, 3, 45, 7, 1, 9]
print("Smallest element:", min(arr))
```

✓ 0.0s Python

Smallest element: 1

Q2. Write a python program to implement bubble sort

```
arr = [5,1,4,2,8]

for i in range(len(arr)):
    for j in range(0, len(arr)-i-1):
        if arr[j] > arr[j+1]:
            arr[j], arr[j+1] = arr[j+1], arr[j]

print("Sorted array:", arr)
```

✓ 0.0s Python

Sorted array: [1, 2, 4, 5, 8]

Q3. Write a python program to multiply two matrices.

```

A = [[1,2],[3,4]]
B = [[5,6],[7,8]]

result = [[0,0],[0,0]]

for i in range(2):
    for j in range(2):
        for k in range(2):
            result[i][j] += A[i][k] * B[k][j]

print("Result:", result)

```

✓ 0.0s

Python

Result: [[19, 22], [43, 50]]

Q4. Write a Python class to find validity of a string of parentheses, '(', ')', '{', '}', '[' and ']'. These brackets must be close in the correct order, for example "()" and "()[{}]" are valid but "[]", "{([)]}" and "{{{" are invalid.

```

class Solution:
    def isValid(self, s):
        stack = []
        pairs = {'(': ')', '[': ']', '{': '}' }

        for ch in s:
            if ch in "({[":
                stack.append(ch)
            else:
                if not stack or stack[-1] != pairs[ch]:
                    return False
                stack.pop()
        return not stack

# s = input("Enter string: ")
# s = "([{}])"
print(Solution().isValid(s))

```

✓ 0.0s

Python

True

Q5. Write a python class to reverse a string word by word.

```

s = input("Enter string: ")
print("Reversed:", " ".join(s.split()[::-1]))

```

✓ 17.3s

Python

Reversed: word by word sentence this reverse lets

Q6. Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.

```
class Circle:
    def __init__(self, r):
        self.r = r

    def area(self):
        return 3.14 * self.r * self.r

    def perimeter(self):
        return 2 * 3.14 * self.r

c = Circle(5)
print("Area:", c.area())
print("Perimeter:", c.perimeter())
```

✓ 0.0s

Python

Area: 78.5

Perimeter: 31.400000000000002