



# Department of Computer Science and Engineering

<b>Course Code: CSE220</b>	<b>Credits: 1.5</b>
<b>Course Name: Data Structure</b>	<b>Semester: Summer '20</b>

## Lab 4

### Recursion

#### I. Topic Overview:

Students will be able to understand recursive method/function calling. They will have hands-on practice in solving a few basic tasks using recursion and later some practical use of recursion will be covered as well.

#### II. Lesson Fit:

The lab itself should be followed by the previous labs and theory knowledge on the recursion and function calling.

#### III. Learning Outcome:

After this lecture, the students will be able to:

- Define a recursive solution for a given task.
- Understand the importance of recursion.
- Practice and solve real-life problems.

#### IV. Anticipated Challenges and Possible Solutions

- Task 1: Students may get confused during repetitive calls of the same method recurrently.

##### **Solutions:**

- i. Visualizing the recursive flow of the entire call will help them to understand.
- b. Task 1: Nested recursion (a recursive function call from the inside of another recursive call) can be confusing.

**Solutions:**

- i. Detail visualization with the calling direction can be used to clear such confusion.

**V. Acceptance and Evaluation**

Students will be evaluated on their submitted tasks.

**VI. Activity Detail**

**a. Hour: 1**

**Tutorial:**

Students must watch the video tutorial updated on BUX lab4's folder

**b. Hour: 2**

**Discussion:**

As the students are supposed to get the basics from the theory videos and lab tutorial, a live session will be held to discuss the problems they faced and provide further explanations if necessary.

## **Lab 4 Activity List**

### **Task 1**

Implement a recursive algorithm to find factorial of  $n$ .

### **Task 2**

Implement a recursive algorithm to find the  $n$ -<sup>th</sup> Fibonacci number.

### **Task 3**

Print all the elements of a given array recursively.

### **Task 4**

Implement a recursive algorithm that takes a decimal number  $n$  and converts  $n$  to its corresponding (you may return as a string) binary number.

### **Task 5**

Implement a recursive algorithm to find the  $m^n$ .

$m = 5$ .  $n = 4$

ans:  $5*5*5*5$

$m*n$

### **Task 6**

Implement a recursive algorithm to add all the elements of a non-dummy headed singly linked linear list. Only head of the list will be given as parameter where you may assume every node can contain only integer as its element.

Note: you'll need a Singly Node class for this code.

### **Task 7**

Implement a recursive algorithm which will print all the elements of a non-dummy headed singly linked linear list in reversed order.

Example: if the linked list contains 10, 20, 30 and 40, the method will print

40

30

20

10

Note: you'll need a Singly Node class for this code.