**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| **Course Code:CSE220** | **Credits: 1.5** |
| **Course Name: Data Structure** | **Semester: Fall’18** |

**Lab 9  
Recursion**

1. **Topic Overview:**

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

1. **Lesson Fit:**

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

1. **Learning Outcome:**

After this lecture, the students will be able to:

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

1. **Anticipated Challenges and Possible Solutions**
   1. Task 1: Students may get confused during repetitive call of the same method recurrently.

**Solutions:**

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

**Solutions**:

* + 1. Detail visualization with the calling direction can be used to clear such confusion.

1. **Acceptance and Evaluation**

Students will be evaluated according to their progress in the lab as they complete each problem. Maybe some of the students will not be able to finish all the tasks; they will submit those later and give an oral justification to get their performance mark.

1. **Activity Detail**
   1. **Hour: 1  
      Discussion:**1. A short quiz on the previous lab topic.  
      2. Evaluating and discussing the quiz question. **Problem Task:**
      1. Quiz question will be prepared by the lab faculty members
   2. **Hour: 2**

**Discussion:**

Give a basic overview of recursion. As the students are supposed to get the basics from the theory class, a brief intro should suffice. Discuss the first three tasks.

**Problem Task:**

* + 1. Task 1 to 3
  1. **Hour: 3**

**Discussion:**

Check their progress on the first three and then discuss the last four tasks.

**Problem Task:**

* + 1. Task 4 to 7

1. **Home tasks**
   1. Any unfinished tasks

**Lab 1 Activity List**

**Task 1**

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

**Task 2**

Implement a recursive algorithm to find the *n*-th 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.

**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.