

## **IOOM - Object Oriented Programming**

### **Lab Assignment – 1**

### **Batch 1& 2**

**Evaluation: Monday August 6, 2018, at 2.00 PM**

#### **Problem 1:**

Define a stack using structure in C. Create functions to push elements into and pop elements off the stack. Also create a function to print out the elements currently present in the stack. In the main function, create a dynamic stack by invoking the stack functions (push and pop).

Each node contains item details with the following structure:

1. Item id (integer).
2. Quantity (integer).
3. Item Label (Character array of size 10).
4. Pointer to the next node.

#### **Problem 2:**

Define a stack using structure in C. Create necessary functions to push, pop, print out the elements in the stack. Consider a stack which consists of different electronics products with their quantities (integer). Sort the stack such that top of the stack contains product with a highest available quantity. After sorting, assume that the middle position product (all quantity) is sold out, and then delete that product from the stack, without using any additional data structure. Print out the stack elements after each operation. In the main function, invoke the required functions. Each node contains product details with the following structure:

1. Product id (integer).
2. Quantity (integer).
3. Item Label (Character array of size 10).
4. Pointer to the next node.

## **IOOM - Object Oriented Programming**

### **Lab Assignment – 1**

### **Batch 3 & 4**

**Evaluation Date & Time: Tuesday August 7, 2018 at 2.00 PM**

#### **Problem 1:**

Define a structure for tree node in C. Write a function to insert elements into the tree based on the values, a function to delete an element from the tree while properly rearranging the nodes left after deletion. Finally write a function (recursive or iterative) to traverse the tree in the ascending order of values present in the nodes. Call these operations in the main function to create and manipulate a BST.

Each node represents aggregate marks of a student with the following structure:

1. Student Name (Character array of size 20).
2. Total\_Marks (integer).
3. Pointer to the left child.
4. Pointer to the right child.

#### **Problem 2:**

Define a structure for tree node in C. Write a function to insert elements into tree, a function to delete an element and rearrange the nodes after deletion. Write a function to find the nodes which are cousins of each other. Two nodes are cousins of each other if they are at same level and have different parent node (Take one node as input and find out its cousin nodes). Also write an iterative function for finding the addition of nodes at the maximum level. Call these functions from main function for these operations on BST. Each node is having the following structure.

```
struct node {  
    int data;  
    node *left, *right;  
} *temp
```