**Logo, company name

Description automatically generated**

**Department of (Computer Science)**

**Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan**

**COMP-112L Data Structure** **& Algorithm Lab**

**Lab Journal**

**Class: BS Computer Science**

**Name: Ahmed Raza**

**Registration No.: B20F0436CS031**

**Semester: 4th**

**Submission Date: 26th April 2022**

**Submitted to: Engr. Rafi-Ullah**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Instructor Signature**

**Lab No. 09**

**Stack**

**Objectives:**

In this lab we will be discussing about Stack in detail. This is one of the most important concepts in C++ language. A stack is an Abstract Data Type (ADT) that used in most programming languages. It is named stack as it behaves like a real-world stack. Stack allows operations at one end only.

**Tools/Software Required:**

* All the tasks are implemented on DEV C++.

**Introduction:**

**Stack**

This feature makes it LIFO data structure.

LIFO stands for **Last-in-first-out**.

We need to provide five operations to implement a stack. These operations are:

* + **isEmpty():** check if stack is empty.
  + **isFull():** check if stack is full.
  + **push():** Pushing (storing) an element on the stack.
  + **pop():** Removing (accessing) an element from the stack.
  + **peek():** Get the top data element of the stack, without removing it.

Here, the element which is placed (inserted or added) last, is accessed first.

**Lab Tasks:**

**Task 1:**

Write a program using c++ to generate a stack1 and push digits from 0 to 9. Then generate another stack that should pop the digits from the stack1 store in varibles and push in stack 2 in such a way that the output should show your enrollment number.

**Code:**

**#include <iostream>**

**using namespace std;**

**int queue[10];**

**int n = 10;**

**int front = - 1;**

**int rear = - 1;**

**int enqueue() {**

**int val;**

**if (rear == n - 1)**

**cout<<"Queue Overflow"<<endl;**

**else {**

**if (front == - 1)**

**front = 0;**

**cout<<"Insert the element in queue : "<<endl;**

**cin>>val;**

**rear++;**

**queue[rear] = val;**

**}**

**}**

**void dequeue() {**

**if (front == - 1 || front > rear) {**

**cout<<"Queue Underflow ";**

**return ;**

**} else {**

**cout<<"\nElement deleted from queue is : "<< queue[front] <<endl;**

**front++;;**

**}**

**}**

**void Display() {**

**if (front == - 1)**

**cout<<"Queue is empty"<<endl;**

**else {**

**cout<<"\n Queue elements are : ";**

**for (int i = front; i <= rear; i++)**

**cout<<queue[i]<<" ";**

**cout<<endl;**

**}**

**}**

**int main(){**

**enqueue();**

**enqueue();**

**enqueue();**

**enqueue();**

**Display();**

**dequeue();**

**dequeue();**

**dequeue();**

**Display();**

**}**

**Output:**

**Text

Description automatically generated**

**Task 2:**

* **Give one examples of stacks found in real life?**

**Real life example of stack:**

* 1: A most popular example of stack is plates in marriage party. Fresh plates are **pushed** onto to the top and **popped** from the top.



2: To reverse a word, You push a given word to stack - letter by letter - and then pop letters from the stack.

3: An *"undo"* mechanism in text editors; this operation is accomplished by keeping all text changes in a stack.

Undo/redo stacks in Excel or Word.

Languages processing:

space for parameters and local variables is created internally using a stack.

compiler's syntax check for matching braces is implemented by using stack.

4: A stack of plates/books in a cupboard.

5: Wearing/Removing Bangles.

6: Support for recursion

Activation records of method calls.

**Results & Observations:**

In this Lab I've learned about the concept of Queue & also understand that how we can insert a node by using Enqueue function and delete by using Dequeue function from the stack. Then by using Display function I’ve print out the remaining node after deleting.