­­



Submitted to

:

Sir Mohsin

***BS***

***-***

***Software Engineering***

***3***

***rd***

***-***

***E***

Title: Assignment 3

DSA

Hamza Mehmood

Roll# SP-21-110

NUML

-

S21

-

2352

9



**National University of Modern Languages**

DATA STRUCTURES & ALGORITHMS

**Subject:** Data Structures and Algorithms **Instructor:** Mohsin Abbas

**Lab Assignment:** 3 **Due Date:** 06-04-2022

**Class:** BSSE III (Eve)

**Student Roll No:SP-21-110**  **Student Name:Hamza Mehmood**

**Total Marks:** 20 **Obtained Marks:**

// Authors:-Hamza Mehmood(SP-21-110).

// Program Title:-Assignment # 3

//Preprocessor Directive Section

#include <iostream>

#define MAX\_SIZE 100

//Standard Allocation

using namespace std;

void Intro() //Function For Introduction

{

    cout << "\t\t    /\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\/\\\n"

         << endl;

    cout << "\t\t      National University of Modern Languages \n"

         << endl;

    cout << "\t\t    \t Software Engineering Department \n"

         << endl;

    cout << "\t\t  \t  Instructor:- Prof.Mohsin Abbas \n"

         << endl;

    cout << "\t\t\t\t--Assignment # 3--    \n"

         << endl;

    cout << "------------------------------------------------------------------------------\n"

         << endl;

    cout << "\t\t |Name| = Hamza Mehmood(SP-21-110)\n "

         << endl;

    cout << "\t\t      |Roll #| = SP-21-110\n "

         << endl;

    cout << " \\\*\\\*\\\*\\\*\\\*\\\*\\\*\\\*\\\*\\\*\\\*\\\*\\\*\\\*\\\*\\\*\\\*\\\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\*/\n"

         << endl;

}

class Queue {

private: //limiting these data members to private so that other class can't access these.

    int item; // variable for Current Position

    int arr\_queue[MAX\_SIZE];// Creating Array of size 100

    int rear; //variable for processing algorithm

    int front;//variable for processing algorithm

    int numtofind;

public:

    Queue() { //Constructor for Initializing Class Elements.

        rear = 0;

        front = 0;

    }

//Insertion Method

    void insert() {

         //When The current position of queue is equal to the total size of queue the it is said that the queue is full.

        if (rear == MAX\_SIZE)

            cout << "\n Queue is FULL";

        else {

            cout << "\nEnter The Value to be Insert : ";

            cin>>item;

            //Position : 1 , Insert Value : Value inserted

            cout << "\n Position : " << rear + 1 << " , Inserted Value  : " << item;

            arr\_queue[rear++] = item;

        }

    }

//Function for Deletion

    void Delete() {

         // Front value is to be determined when a value is deleted from queue and rear is the current position of top current value.if both these are equal the the queue is empty.

        if (front == rear)

            cout << "\n Queue is Empty!";

        else {

            cout << "\n Position : " << front << " , Removed Value  :" << arr\_queue[front];

            front++;

        }

    }

    void display() {

        cout << "\n Queue Size : " << (rear - front);

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

            cout << "\n Position : " << i << " , Value  : " << arr\_queue[i];

    }

    void search(){

        bool check = false; //Initializing Decision Variable

        cout << "\n";

        cout << "Enter Value to be Found = ";

        cin >> numtofind;

        for (int i = 0; i < MAX\_SIZE; i++)

        { // loop will be implemented for every Iteration of i and will be break if the value of that such ith iteration is equal to the num to find.

            if (arr\_queue[i] == numtofind)

            {

                check = true;

                break;

            }

        }

        if (check == false)

        {

            cout << "Value Not Found";

        }

        else

        {

            cout << "Value Found";

        }

}

};

int main()

{

    Intro();

    cout << "\n\n";

    cout << "Question 1. Write a program that implements queue data structure using class based implementation.Provide menu for user to select operations for INSERTION,DELETION,DISPLAY and make Queue Empty.Take value in queue from user.Add the search operation using linear search algorithm which enables user to search a value from queue.";

    cout << "\n\n";

    int choice, exit\_p = 1;

    //Creating Object of Class

    Queue obj;

    cout << "\nSimple Queue Example - Class and Member Functions in C++";

    // Implementing a do-while Loop for the Execution of Program.

    do {

        cout << "\n\n Queue Main Menu";

        cout << "\n1.Insert \n2.Remove \n3.Display \n4.Search  \nOthers to exit";

        cout << "\nEnter Your Choice : ";

        //asking for user choice

        cin>>choice;

        //Determining Choice

        switch (choice) {

            case 1:

                obj.insert();

                break;

            case 2:

                obj.Delete();

                break;

            case 3:

                obj.display();

                break;

            case 4:

                obj.search();

                break;

            default:

                exit\_p = 0;

                break;

        }

    } while (exit\_p);

    return 0;}

Screen Shots







