|  |
| --- |
| **Computer Engineering Department - ITU** |
| **CE200L: Data Structures & Algorithms Lab** |

|  |  |
| --- | --- |
| **Course Instructor: Usama Bin Shakeel** | **Dated: 04/10/2022** |
| **Teaching Assistant: Muhammad Sufyan Ashraf** | **Semester: Fall 2022** |
| **Lab Engineer: Nadir Abbas** | **Batch: BSCE2021** |

# **Lab 6B. Queues with Arrays**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Roll number** | **Report**  **(out of 100)** | **Scaled to 10** | **Total**  **(out of 10)** |
| NIMRA MAQBOOL | BSCE21012 |  |  |  |

Checked on: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## **Objective**

The objective of this lab is to provide the knowledge of basic data structures and their implementations.

## **Equipment and Component**

|  |  |  |
| --- | --- | --- |
| **Component Description** | **Value** | **Quantity** |
| Computer | Available in lab | 1 |

## **Conduct of Lab**

1. Students are required to perform this experiment individually.
2. In case the lab experiment is not understood, the students are advised to seek help from the course instructor, lab engineers, assigned teaching assistants (TA) and lab attendants.

## **Theory and Background**

A **queue** is defined as a linear data structure that is open at both ends and the operations are performed in First In First Out (FIFO) order. We define a queue to be a list in which all additions to the list are made at one end, and all deletions from the list are made at the other end.

**Templates** are a feature of the C++ programming language that allows functions and classes to operate with generic types. This allows a function or class to work on many different data types without being rewritten for each one.

**Lab Task**

**Task A**

**// Add function to insert the new element in queue**

void enqueue ()

{

}

**// Add function to remove the element at front from queue**

void dequeue ()

{

}

**// Add function to show the element at front**

void showFront ()

{

}

**// Add function to check if queue is empty**

void isEmpty ()

{

}

|  |
| --- |
| // Paste your code here  **FUNCTION.H:**  #include <iostream> using namespace std; template<typename T>  class queue { public:  T size;  T count;  T iFront=0;  T iBack=-1; //declaring public members  T \*array;   queue(T s) {  size = s; //making a constructor to set the values  count = 0;  }  bool isEmpty() {  if (iFront == 0 && iBack == -1 && count== 0 || count==size ) { //making a function to check that is the queue is empty or not  cout << "ARRAY IS EMPTY." << endl;  return true;  }  else{  cout<<"ARRAY IS NOT EMPTY."<<endl;  return false;  }  }   void showFront() {  if(isEmpty()) {  cout << " "; //checking if the queue is empty  }else {   cout << "THE ELEMENT IS = " << array[iFront] << " "; //showing the index of iFront  iFront++; //iterating the iFront  }  }   bool isFull() {  if (iFront == 0 && iBack == size || count==size) { //checking if the queue is full or not  cout << "ARRAY IS NOT EMPTY." << endl;  return true;  }  else{  cout<<"THE ARRAY IS NOT FULL."<<endl;  return false;  }  }  void enqueue(T value) {  if (isFull()) {  cout<<" "; //if the queue is full  }  else{  iBack++; //iterating  iBack = (iBack+1) % size; //taking mode  array[iBack] = value; //putting the value  cout<<"THE ADDED VALUE = "<<array[iBack]<<endl; //displaying  size++; //incrementing size  count++; //incrementing count  }  }   void dequeue() {  if(isEmpty()){  cout<<" "; //checking if the queue is empty or not  }  else{  if(count==size){ //if the count is equal to size  cout<<"THE ARRAY'S COUNT IS AT THE END."<<endl; //display this message  cout<<"SO THE VALUE IS ZERO."<<endl;  }  cout<<"VALUE REMOVED = "<<array[iFront]<<endl; //displaying  iFront++; //iterating front pointer  iFront = (iFront+1) % size; //taking mode  size--; //decrementing size  count--; //decrementing count  }   }   };  **MAIN.CPP:**  #include <iostream> #include "Functions.h"  using namespace std;  int main() {  queue<int> q(5);  int opt;  int opt1;  do {  cout << "CHOOSE OPTION." << endl;  cout << "1.SHOW FRONT." << endl;  cout << "2.ENQUEUE." << endl;  cout << "3.DEQUEUE." << endl;  cout << "4.EXIT." << endl;  cin >> opt;  if (opt == 1) {  q.showFront();  }  if (opt == 2) {  do {  cout << "YOU WANT TO ENTER NUMBER IN QUEUE?" << "\nENTER 1 FOR YES AND 0 FOR" << endl;  cin >> opt1;  if (opt1 == 1) {  int value;  cout << "ENTER THE VALUE TO ADD IN THE LIST." << endl;  cin >> value;  q.enqueue(value);  }  if (opt == 0) {  cout << "YOU CHOOSE NO." << endl;  exit(3);  }  } while (opt1 > 0 && opt1 <= 1);   }  if (opt == 3) {  do {  cout << "YOU WANT TO DELETE NUMBER IN QUEUE?" << "\nENTER 1 FOR YES AND 0 FOR NO" << endl;  cin >> opt1;  if (opt1 == 1) {  q.dequeue();  }  if (opt == 0) {  cout << "YOU CHOOSE NO." << endl;  break;  }  } while (opt1 > 0 && opt1 <= 1);  }  if (opt == 4) {  cout << "YOU CHOOSE TO EXIT." << endl;  exit(3);  }  } while (opt >= 1 && opt <= 4); }  // Paste your output here  **OUTPUT:**    **Text  Description automatically generated**  Text  Description automatically generated |

#### **Assessment Rubric for Lab**

**Method for assessment:**

Lab reports and instructor observation during lab sessions. Outcome assessed:

a. Ability to conduct experiments, as well as to analyze and interpret data (P) b. Ability to function on multi-disciplinary teams (A)

c. Ability to use the techniques, skills, and modern engineering tools necessary for engineering practice (P)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Performance metric** | **Task** | **CLO** | **Description** | **Max marks** | **Exceeds expectation** | **Meets expectation** | **Does not meet expectation** | **Obtained marks** |
| 1. Realization of experiment (a) | 1 | 1 | Functionality | 40 | Executes without errors excellent user prompts, good use of symbols, spacing in output. Through testing has been completed (35-40) | Executes without errors, user prompts are understandable, minimum use of symbols or spacing in output. Some testing has been completed (20-34) | Does not execute due to syntax errors, runtime errors, user prompts are misleading or non-existent. No testing has been completed (0-19) |  |
| 2. Teamwork (b) | 1 | 3 | Group Performance | 5 | Actively engages and cooperates with other group member(s) in effective manner (4-5) | Cooperates with other group member(s) in a reasonable manner but conduct can be improved (2-3) | Distracts or discourages other group members from conducting the experiment (0-1) |  |
| 3. Conducting experiment (a, c) | 1 | 1 | On Spot Changes | 10 | Able to make changes (8-10) | Partially able to make changes (5-7) | Unable to make changes (0-4) |  |
| 1 | 1 | Viva | 10 | Answered all questions (8-10) | Few incorrect answers (5-7) | Unable to answer all questions (0-4) |  |
| 4. Laboratory safety and disciplinary rules (a) | 1 | 3 | Code commenting | 5 | Comments are added and does help the reader to understand the code (4-5) | Comments are added and does not help the reader to understand the code (2-3) | Comments are not added (0-1) |  |
| 5. Data collection (c) | 1 | 3 | Code Structure | 5 | Excellent use of white space, creatively organized work, excellent use of variables and constants, correct identifiers for constants, No line-wrap (4-5) | Includes name, and assignment, white space makes the program fairly easy to read. Title, organized work, good use of variables (2-3) | Poor use of white space (indentation, blank lines) making code hard to read, disorganized and messy (0-1) |  |
| 6. Data analysis (a, c) | 1 | 4 | Algorithm | 20 | Solution is efficient, easy to understand, and maintain (15-20) | A logical solution that is easy to follow but it is not the most efficient (6-14) | A difficult and inefficient solution (0-5) |  |
| 7. Computer use (c) | 1 | 2 | Documentation & Github Submissions | 5 | Timely (4-5) | Late (2-3) | Not done (0-1) |  |
|  | Max Marks (total): | | | 100 | Obtained Marks (total): | | |  |

Lab Engineer Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_