**National Textile University,**

**Faisalabad**



**Department of Computer Science**

|  |  |
| --- | --- |
| **Name:** | Asadullah |
| **Class:** | BSCS(A) |
| **Registration No:** | 22-NTU-CS-1150 |
| **Lab:** | Final Project |
| **Course Code:** | COC-2073 |
| **Course Name:** | Data Structure |
| **Submitted To:** | *Mr.Abdul Basit* |

**Proposal for Data Structure and Algorithm Project**

**Title: Tool Cabinet**

**Leader: Asad 22-NTU-CS-1150**

**Overview:**

The Tool Cabinet is a console-based application designed to make easy the understanding and implementation of Data Structures and Algorithms concepts. The project provides a user-friendly menu-driven interface that allows user to explore and interact with some DSA topics. It covers the main concepts of DSA.

**Objectives:**

* Understand and apply various data structures in project.
* Develop a console application in C++ that simulates key concepts of Data Structure.
* Enhance problem-solving and programming skills in a real-world application context.

**Functionalities:**

* **Main menu**

Display a list of DSA topic (Queue, Tree, Heap)

User input to navigate with topics.

* **Sub-menu**

Show a list of operations for selected topic.

User give input to navigate with operation

* **Queue**

Display a list of Queue Operation and user navigate with his choice.

* **Binary Search Tree**

Display a list with Tree Operation and user navigate with choice.

* **Heap**

Display a list with heap Operation and user navigate with choice.

**Tools:**

* **Visual Studio**

A code-editor that would be use in code implementation of the Tool Cabinet.

**Conclusion:**

Tool Cabinet is a friendly User-interface environment that will demonstrate key concepts of various data structure. It provides a menu for exploring Queue, Binary Search Tree, and Heaps. Users can interact with their chosen data structure, allowing them to create and manipulate Queue, Binary Search Tree, and Heaps.

**Heap:**

I make both Max Heap and Min Heap. We can manipulate both type of heap and can also perform their all functionalities. It is very helpful to understand and demonstrate the core concept of Heap.

**Functionalities:**

cout << "-----Enter 1 for Max heap-----";

cout << "-----Enter 2 for Min heap-----";

cout << "-----Enter 3 to Exits from Heap-----";

* + **Max-Heap**

Max-heap will provide the following operations within friendly menu-base interface that a user can easily perform…

**Functionalities:**

cout << "Enter 1 for Insertion into Max Heap\n";

cout << "Enter 2 for Searching from Max Heap\n";

cout << "Enter 3 for Deletion of Top Eelment from Max Heap\n";

cout << "Enter 4 To Display Max Heap\n";

cout << "Enter 5 To Display Data Max\_Heap File\n";

cout << "Enter 0 To Out from Max Heap\n";

* + **Min-Heap**

Min-heap will also provide the following operations within friendly menu-base interface that a user can easily perform…

**Functionalities:**

cout << "\t------- MIN HEAP OPERATION --------\n\n";

cout << "Enter 1 for Insertion into Min Heap\n";

cout << "Enter 2 for Searching from Min Heap\n";

cout << "Enter 3 for Deletion of Top Eelment from Min Heap\n";

cout << "Enter 4 To Display Min Heap\n";

cout << "Enter 5 To Display Data Min\_Heap File\n";

cout << "Enter 0 To Out from Min Heap\n";

**Queue:**

It is FIFO data structure, here we demonstrate the deep concepts of Queue. All basic are operation fully implemented that will be very helpful for students. It will provide menu-base interface for user:

cout << "--------- Enter 1 To Manipulate With Queue --------- ";

cout << "--------- Enter 2 To Manipulate With Heap --------- ";

cout << "--------- Enter 3 To Manipulate With BST --------- ";

**Functionalities:**

cout << "\t------- QUEUE OPERATION --------";

cout << "Enter 1 For Inserting into Queue";

cout << "Enter 2 To Delete Top Element of Queue";

cout << "Enter 3 to See Top Element of Queue";

cout << "Enter 4 to Dispaly Queue";

cout << "Enter 0 to Exits from Queue";

**Binary Search Tree:**

BST is a non-linear data structure. Here we demonstrate it with his functionalities. It also provide attractive menu-base interface.

cout << "--------- Enter 1 To Manipulate With Queue --------- ";

cout << "--------- Enter 2 To Manipulate With Heap --------- ";

cout << "--------- Enter 3 To Manipulate With BST --------- ";

**Functionalities:**

cout << "\t------- Binary Search Tree OPERATION --------";

cout << "Enter 1 For Inserting into BST";

cout << "Enter 2 To Deletion from BST";

cout << "Enter 3 to Find Maximum Number";

cout << "Enter 4 to Find Minimum Number";

cout << "Enter 5 to Traversal BST";

cout << "Enter 0 to Exits from BST";