

Activity No. 5.1	
Hands-on Activity 5.1 Queues	
Course Code: CPE010	Program: Computer Engineering
Course Title: Data Structures and Algorithms	Date Performed: 10/07/2024
Section: CPE21S1	Date Submitted: 10/08/2024
Name(s): Jhon Hendricks T. Bautista	Instructor: Mrs. Maria Rizette Sayo

6. Output

main.cpp

Share

Run

```
1 #include <iostream>
2 #include <queue>
3 #include <string>
4 using namespace std;
5
6 void display(queue<string> student_q) {
7     while(!student_q.empty()){
8         cout<< "My name is " << student_q.front() << endl;
9         student_q.pop();
10    };
11 }
12
13 int main() {
14     queue<string> student_q;
15
16     string students [4] = {"Kurt", "Dale", "Redj", "Jhon"};
17
18     for(int i = 0; i < size(students); i++){
19         string temp = students[i];
20         student_q.push(temp);
21     };
22
23     display(student_q);
24
25     return 0;
26 }
27
28
```

/tmp/bKHLZsoXYF.o

My name is Kurt
My name is Dale
My name is Redj
My name is Jhon

=== Code Execution Successful ===

Table 5-1. Queues using C++ STL

main.cpp	Output
<pre> 38 head = head -> next; 39 delete current; 40 } 41 42 int main() { 43 Node* head = new Node(); 44 string name; 45 // first item in the queue 46 cout << "Enter the first name to insert in the queue: "; 47 getline(cin, name); 48 head -> name = name; 49 head -> next = NULL; 50 51 traversal(head); 52 53 // inserting items into queue 54 cout<<"\n"; 55 insert_item("Kurt", head); 56 insert_item("Redj", head); 57 traversal(head); 58 59 cout<<"\n"; </pre>	<pre> /tmp/8yTqjHkDK0.o Enter the first name to insert in the queue: Hendricks Students Hendricks Students Hendricks Kurt Redj After deletion Students Kurt Redj === Code Execution Successful === </pre>

Table 5-2. Queues using Linked List Implementation

main.cpp	Output
<pre> 59 }; 60 61 int main() { 62 int maxSize; 63 cout << "Enter the maximum size of the queue: "; 64 cin >> maxSize; 65 cin.ignore(); 66 67 Queue queue(maxSize); 68 string name; 69 70 cout << "Enter the first name to insert in the queue: "; 71 getline(cin, name); 72 queue.enqueue(name); 73 74 queue.enqueue("Dale"); 75 queue.enqueue("Kurt"); 76 queue.enqueue("Redj"); 77 78 queue.display(); </pre>	<pre> /tmp/1r4Bjo434S.o Enter the maximum size of the queue: 5 Enter the first name to insert in the queue: hendricks Students: hendricks Dale Kurt Redj After using dequeue Students: Dale Kurt Redj === Code Execution Successful === </pre>

Table 5-3. Queues using Array Implementation

7. Supplementary Activity

```
/tmp/FnrLS6rbBE.o
```

```
Job added to queue: Job ID: 1, User: Hendricks, Pages: 10
```

```
Job added to queue: Job ID: 2, User: Redj, Pages: 5
```

```
Job added to queue: Job ID: 3, User: Kurt, Pages: 15
```

```
Job added to queue: Job ID: 4, User: Dale, Pages: 3
```

Processing Jobs

```
Processing Job ID: 1, User: Hendricks, Pages: 10
```

```
Processing Job ID: 2, User: Redj, Pages: 5
```

```
Processing Job ID: 3, User: Kurt, Pages: 15
```

```
Processing Job ID: 4, User: Dale, Pages: 3
```

```
All jobs processed
```

For me I used the linked list implementation of the queue because it provided me an easier time to build the program. Also the insertion of data and manipulation of the queue was easier to do because of the use of pointers. There was no memory wasted or unallocated since the linked list uses a dynamic allocation. This is an advantage because we do not know how big the queue will get. A linked list provides the optimum flexibility and efficiency for the printer queue scenario, where jobs are added sequentially and processed in a first-come, first-served order, without the disadvantages of resizing or shifting elements as in an array implementation.

8. Conclusion

In this activity, I was able to know the basic structure of the implementation of the queue in a C++ program. I was able to understand the 3 different ways we can implement a queue which is the use of STL C++, linked list, and array. I followed carefully the explanation of the different methods like dequeue, enqueue, and to always remember that a queue is "First In, First Out". Then I was able to use my learnings in a simple program about a printer simulation. I can conclude that queue is an important data structure made for certain scenarios where FIFO principle is needed to make the work efficient.

9. Assessment Rubric