OOP Lab Assignment 7

Jaynam Modi. PG-43. G3. Sep 19, 2020

- Problem Statement : Write C++ program using STL for implementation of stack & queue using List Container
- Data Members : Integer Values
- Member Functions :
  - Create STL stack and Queue of Integer values
  - Insert items in stack and queue
  - Remove item form stack and queue
  - Return top most element of stack
  - Return the front and back element of stack
  - Return the number of elements in stack and queue
- Objectives :

1. To learn the concepts of STL in C++ that can be used to implement data structures and algorithms

2. To learn the use and working of stack and queue of STL

3. To learn insert, delete, and return data items in stack and queue.

- Theory :

The Standard Template Library (STL) is a set of C++ template classes to provide common programming data structures and functions such as lists, stacks, arrays, etc. It is a library of container classes, algorithms, and iterators. It is a generalized library and so, its components are parameterized.

Stack container in STL is a type of container adaptors. It is used to replicate a stack data structure in C++. Stack container is a set of elements in which the elements are inserted at one end and are also deleted at the same end.

The queue is yet another container in STL which is very simple and useful too. Queue container is a replica of the queue data structure in C++. Unlike stack, in the queue container, there are two ends, i.e. front, and back.

Elements are added to the queue at the back while deleted from the front of the queue. In general, queue uses FIFO (First in, First Out) type of arrangement.

Operations :

- push : push operation is used to insert an element in the stack. This operation always adds elements at the top of the stack.

- pop: pop operation is used to remove an element from the stack. The element removed is the one that is pointed to by the top of the stack. As a result of the pop operation, the stack size is reduced by 1.

- empty: Checks if the stack is empty or not.

Algorithm :

1. Initialise

2. Perform user defined operations on stack

3. Exit

- Platform :

    - 64-bit Open source Linux or its dderivatives.

    - Open Source C++ Programming tool like G++

- Input : Numbers

- Output :

1. Removed item form stack and queue using pop( ) function

2. Returned top most element of stack

3. Returned the front and back element of stack

4. Number of elements in stack and queue

- Conclusion : Thus, implemented the stack and queue using STL. This System is able insert, delete, and return the data items from stack and queue.

- FAQs :

1. What is STL?

> The Standard Template Library (STL) is a set of C++ template classes to provide common programming data structures and functions such as lists, stacks, arrays, etc. It is a library of container classes, algorithms, and iterators. It is a generalized library and so, its components are parameterized.

2. What is container? Explain different types of containers

> A container is a holder object that stores a collection of other objects (its elements). They are implemented as class templates, which allows a great flexibility in the types supported as elements.

> Stack container in STL is a type of container adaptors. It is used to replicate a stack data structure in C++. Stack container is a set of elements in which the elements

are inserted at one end and are also deleted at the same end.

> The queue is yet another container in STL which is very simple and useful too. Queue container is a replica of the queue data structure in C++. Unlike stack, in the queue container, there are two ends, i.e. front, and back.

3. Explain stack and queue?

> Stack container in STL is a type of container adaptors. It is used to replicate a stack data structure in C++. Stack container is a set of elements in which the elements are inserted at one end and are also deleted at the same end.

> The queue is yet another container in STL which is very simple and useful too. Queue container is a replica of the queue data structure in C++. Unlike stack, in the queue container, there are two ends, i.e. front, and back.

4. Which are different member functions of stack container?

> push : push operation is used to insert an element in the stack. This operation always

adds elements at the top of the stack.

- pop: pop operation is used to remove an element from the stack. The element removed is the one that is pointed to by the top of the stack. As a result of the pop operation, the stack size is reduced by 1.

- top: Returns the topmost element of the stack.

- empty: Checks if the stack is empty or not.

- size: Returns the size of the stack i.e. the number of elements in the stack.

5. Which are different member functions of queue container?

- push: Function ‘push’ adds the element at the end of the queue i.e. at the back of the queue.

- pop: Function ‘pop’ removes the first element of the queue i.e. the element at the front of the queue.

- front: This function returns a reference to the first element of the queue.

> back: Back returns a reference to the last element in the queue.

> empty: Checks if the queue is empty.

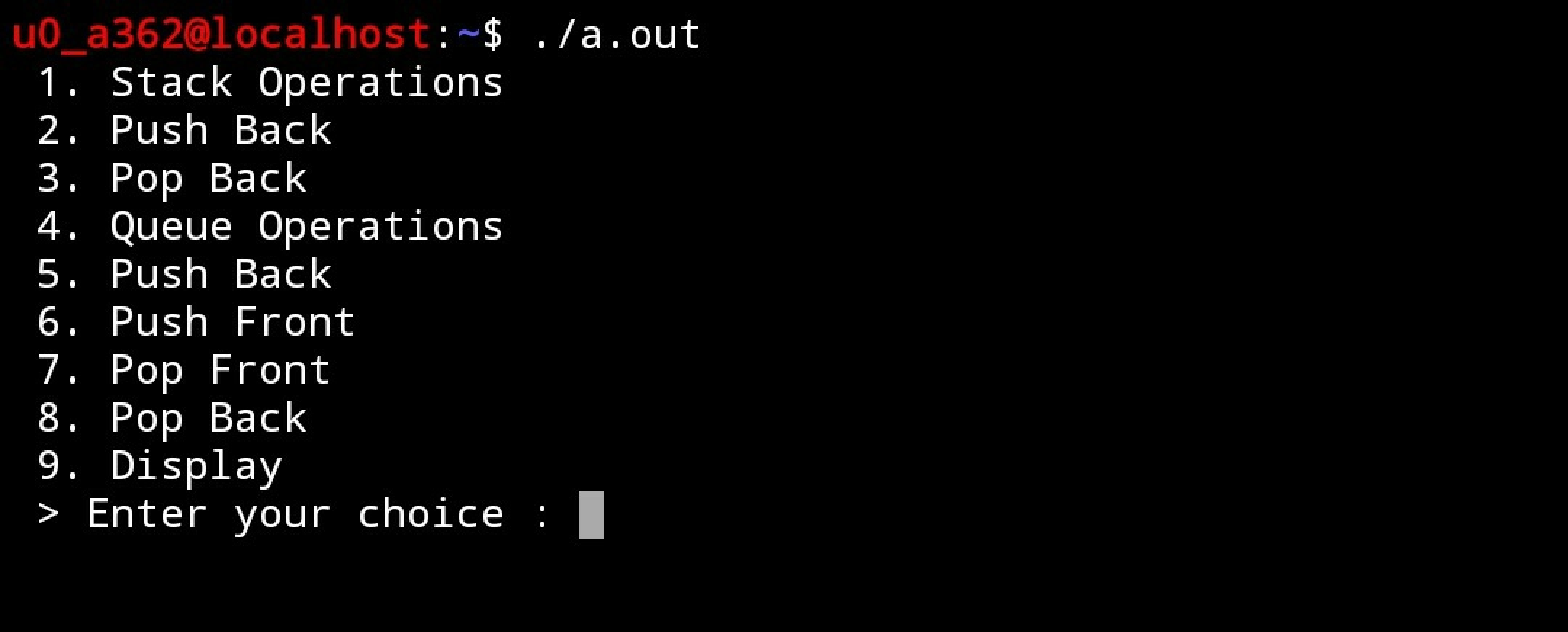
> size: Returns the size of the queue i.e. the number of elements in the queue.

```cpp
#include <iostream>
#include <cstdlib>
#include <list>
#include <iterator>

using namespace std;

template <class T>
class container{
    public: T item,a,b;
 };


int main(){
  int ch;
  container <int>c;
  list<int> g;
  list <int> :: iterator it;
  do{
  cout << " 1. Stack Operations\n 2. Push Back\n 3. Pop Back \n 4. Queue Operations\n 5. Push Back\n 6. Push Front\n 7. Pop Front\n 8. Pop Back\n 9. Display"  <<  endl;
  cout << " > Enter your choice : ";
  cin >> ch;
  switch(ch)
  {
      case 2:cout << "Enter value to be inserted: ";
             cin >> c.item;
             g.push_back(c.item);
             break;
      case 3:  g.pop_back();
      break;
      case 4: cout << " > Queue Operation";
      case 5: cout << " > Enter value to be inserted: ";
             cin >> c.item;
             g.push_back(c.item);
             break;
      case 6: cout << " > Enter value to be inserted: ";
             cin >> c.item;
             g.push_front(c.item);
             break;
        case 7:  g.pop_front();          break;
      case 8:  g.pop_back();          break;
      case 9: cout << " > Elements are";
              for(it = g.begin(); it != g.end(); ++it)
                cout  <<  '\t'  <<  *it;
                cout  <<  '\n';
              break;
  }
  }while(ch<=9);
  return 0;
}
```



```
u0_a362@localhost:~$ ./a.out
 1. Stack Operations
 2. Push Back
 3. Pop Back
 4. Queue Operations
 5. Push Back
 6. Push Front
 7. Pop Front
 8. Pop Back
 9. Display
 > Enter your choice : 
```