Searching

0.2.0

Generated by Doxygen 1.8.17

1 Class Index 1
1.1 Class List 1

2 File Index 3
2.1 File List 3

3 Class Documentation 5
3.1 myFIFO Class Reference 5
3.1.1 Detailed Description 5
3.1.2 Constructor & Destructor Documentation 5
3.1.2.1 myFIFO() 5
3.1.3 Member Function Documentation 6
3.1.3.1 add() 6
3.1.3.2 bufLen() 6
3.1.3.3 getElement() 6
3.1.3.4 lenFull() 7
3.1.3.5 printStats() 7
3.1.3.6 remove() 8
3.2 seqSearch Class Reference 8
3.2.1 Detailed Description 9
3.2.2 Constructor & Destructor Documentation 9
3.2.2.1 seqSearch() 9
3.2.3 Member Function Documentation 9
3.2.3.1 fillStorage() 9
3.2.3.2 printStorage() 10
3.2.3.3 search() 10
3.2.4 Member Data Documentation 10
3.2.4.1 storage 10

4 File Documentation 11
4.1 /home/drseth/CPTR227/20210217SearchClassDemo/src/fifo.cpp File Reference 11
4.1.1 Detailed Description 11
4.2 /home/drseth/CPTR227/20210217SearchClassDemo/src/fifo.h File Reference 12
4.2.1 Detailed Description 13
4.3 /home/drseth/CPTR227/20210217SearchClassDemo/src/main.cpp File Reference 13
4.3.1 Detailed Description 14
4.3.2 Function Documentation 14
4.3.2.1 main() 14

Index 15

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

myFIFO 5
seqSearch 8

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

/home/drseth/CPTR227/20210217SearchClassDemo/src/fifo.cpp
This is a simple implementation of a FIFO queue 11
/home/drseth/CPTR227/20210217SearchClassDemo/src/fifo.h
This is a simple implementation of a FIFO queue 12
/home/drseth/CPTR227/20210217SearchClassDemo/src/main.cpp
This demonstrates header files, separate cpp files, and some searching 13

Chapter 3

Class Documentation

3.1 myFIFO Class Reference

```
#include <fifo.h>
```

Public Member Functions

- myFIFO ()
- bool add (int x)
- int remove ()
- void printStats ()
- int lenFull ()
- int bufLen ()
- int getElement (int ii)

3.1.1 Detailed Description

Implements an integer FIFO

Definition at line 18 of file fifo.h.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 myFIFO()

```
myFIFO::myFIFO ( )
```

Constructor

Definition at line 17 of file fifo.cpp.

```
17                 {
18     for(int ii = 0; ii < bufLength; ii++) {
19         buffer[ii] = 0;
20     }
21 }
```

3.1.3 Member Function Documentation

3.1.3.1 add()

```
bool myFIFO::add (
               int x )
```

Adds a integer to the back of the FIFO

Parameters

x	Integer to add to the FIFO

Returns

true if successful, false otherwise

Definition at line 29 of file fifo.cpp.

```
29                       {
30     //if(bufBack < bufLength) {
31     if(length < bufLength) {
32         buffer[bufBack] = x; // Add value to buffer
33         bufBack++; // equivalent to bufBack = bufBack + 1
34         bufBack = bufBack % bufLength; // wraps around to the beginning
35         length++; // increment length since an element was added
36         return(true);
37     } else {
38         cout « "bufBack exceeded buffer length" « endl;
39         return(false);
40     }
41 }
```

3.1.3.2 bufLen()

```
int myFIFO::bufLen ( )
```

Returns the length of the buffer

Definition at line 101 of file fifo.cpp.

```
101                    {
102     return(bufLength);
103 }
```

3.1.3.3 getElement()

```
int myFIFO::getElement (
               int ii )
```

Returns iith element of the FIFO

Parameters

ii	- which element to return

Definition at line 110 of file fifo.cpp.

```
                            {
    // check ii for invalid values
    // return the iith element
    return(buffer[(bufFront + ii) % bufLength]);
}
```

3.1.3.4 lenFull()

```
int myFIFO::lenFull ( )
```

Returns the number of full spaces in the fifo

Definition at line 94 of file fifo.cpp.

```
                    {
    return(length);
}
```

3.1.3.5 printStats()

```
void myFIFO::printStats ( )
```

Prints the information about the buffer

Definition at line 65 of file fifo.cpp.

```
                        {
    cout « "------------------" « endl;
    cout « "bufFront = " « bufFront « " stored at " « &bufFront « endl;
    cout « "bufBack = " « bufBack « " stored at " « &bufBack « endl;
//             cout « "buffer stored at " « buffer « " is:" « endl;
    cout « "length = " « length « endl;
    // print front
    for(int ii = 0; ii < bufLength; ii++) {
        if(ii == bufFront)
            cout « 'f';
        cout « '\t';
    }
    cout « endl;
    for(int ii = 0; ii < bufLength; ii++) {
        cout « buffer[ii] « '\t';
    }
    cout « endl;
    for(int ii = 0; ii < bufLength; ii++) {
        if(ii == bufBack)
            cout « 'b';
        cout « '\t';
    }
    cout « endl;
    cout « "==================" « endl;
}
```

3.1.3.6 remove()

```
int myFIFO::remove ( )
```

Removes an integer from front of the FIFO

Returns

value removed from FIFO, -999999999 if error

Definition at line 48 of file fifo.cpp.

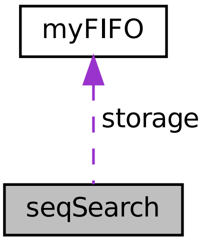
```
48                    {
49      //if(bufFront < bufLength) {
50      if(length > 0) { // bufFront == bufBack means the buffer is empty
51          int retVal = buffer[bufFront];
52          bufFront++;
53          bufFront = bufFront % bufLength;
54          length--; // decrement length since an element was removed
55          return(retVal);
56      } else {
57          cout « "Error tried to remove beyond end of buffer" « endl;
58          return(-999999999);
59      }
60 }
```

The documentation for this class was generated from the following files:

- /home/drseth/CPTR227/20210217SearchClassDemo/src/fifo.h
- /home/drseth/CPTR227/20210217SearchClassDemo/src/fifo.cpp

3.2 seqSearch Class Reference

Collaboration diagram for seqSearch:



Public Member Functions

- seqSearch ()
- void fillStorage (int start)
- void printStorage ()
- int search (int searchTerm)

Public Attributes

- myFIFO storage

  Variable that stores the array.

3.2.1 Detailed Description

Definition at line 14 of file main.cpp.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 seqSearch()

```
seqSearch::seqSearch ( )  [inline]
```

Constructor

Definition at line 21 of file main.cpp.

```
21                  {
22          cout « "Added a seqSearch instance" « endl;
23      }
```

3.2.3 Member Function Documentation

3.2.3.1 fillStorage()

```
void seqSearch::fillStorage (
              int start )  [inline]
```

Fills storage with sequential numbers starting with start

Parameters

start	- The number to start filling at

Definition at line 30 of file main.cpp.

```
30                                 {
31          for(int ii = 0; ii < storage.bufLen(); ii++){
32              storage.add(start++);
33          }
34      }
```

3.2.3.2 printStorage()

```
void seqSearch::printStorage ( )  [inline]
```

Definition at line 36 of file main.cpp.

```
36                        {
37          storage.printStats();
38      }
```

3.2.3.3 search()

```
int seqSearch::search (
            int searchTerm )  [inline]
```

Searches for the value passed

Parameters

searchTerm	

Returns

Returns the location of searchTerm or -1 if not found

Definition at line 46 of file main.cpp.

```
46                              {
47          for(int ii = 0; ii < storage.lenFull(); ii++) {
48
49          }
50          return(-1);
51      }
```

3.2.4 Member Data Documentation

3.2.4.1 storage

```
myFIFO seqSearch::storage
```

Variable that stores the array.

Definition at line 16 of file main.cpp.

The documentation for this class was generated from the following file:

- /home/drseth/CPTR227/20210217SearchClassDemo/src/main.cpp

Chapter 4

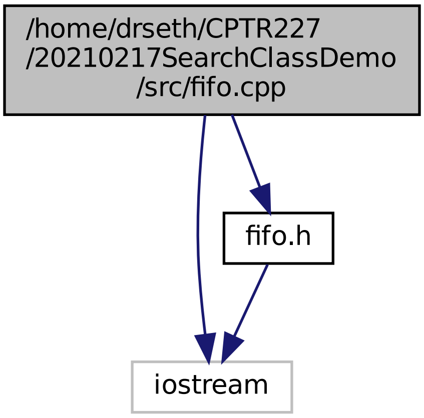
File Documentation

4.1 /home/drseth/CPTR227/20210217SearchClassDemo/src/fifo.cpp File Reference

This is a simple implementation of a FIFO queue.

```
#include <iostream>
#include "fifo.h"
```

Include dependency graph for fifo.cpp:



4.1.1 Detailed Description

This is a simple implementation of a FIFO queue.

This only uses arrays, no STL

Author

Seth McNeill

Date

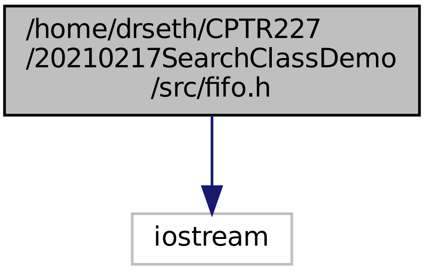
2021 February 02

4.2 /home/drseth/CPTR227/20210217SearchClassDemo/src/fifo.h File Reference

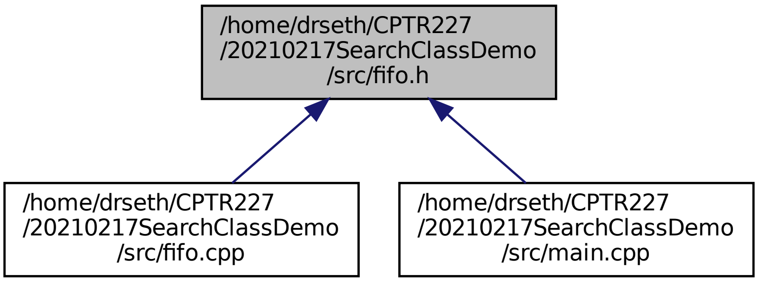
This is a simple implementation of a FIFO queue.

```
#include <iostream>
```

Include dependency graph for fifo.h:



This graph shows which files directly or indirectly include this file:



Classes

- class myFIFO

4.2.1 Detailed Description

This is a simple implementation of a FIFO queue.

This only uses arrays, no STL

Author

Seth McNeill

Date

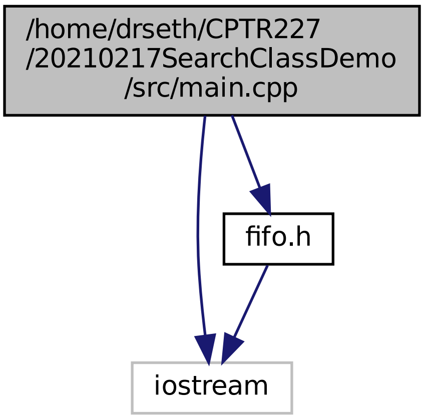
2021 February 02

4.3 /home/drseth/CPTR227/20210217SearchClassDemo/src/main.cpp File Reference

This demonstrates header files, separate cpp files, and some searching.

```
#include <iostream>
#include "fifo.h"
```

Include dependency graph for main.cpp:



Classes

- class seqSearch

Functions

- int main (int, char ∗∗)

4.3.1 Detailed Description

This demonstrates header files, separate cpp files, and some searching.

Implements and times sequential searching using FIFO class

Author

Seth McNeill

Date

2021 February 17

4.3.2 Function Documentation

4.3.2.1 main()

```
int main (
             int ,
             char **  )
```

Definition at line 55 of file main.cpp.

```
55                      {
56     seqSearch s1;
57     s1.fillStorage(0);
58     s1.printStorage();
59     cout « "Search for 1 returns " « s1.search(1) « endl;
60 }
```

Index

/home/drseth/CPTR227/20210217SearchClassDemo/src/fifo.cpp, 11
/home/drseth/CPTR227/20210217SearchClassDemo/src/fifo.h, 12
/home/drseth/CPTR227/20210217SearchClassDemo/src/main.cpp, 13

add
    myFIFO, 6

bufLen
    myFIFO, 6

fillStorage
    seqSearch, 9

getElement
    myFIFO, 6

lenFull
    myFIFO, 7

main
    main.cpp, 14
main.cpp
    main, 14
myFIFO, 5
    add, 6
    bufLen, 6
    getElement, 6
    lenFull, 7
    myFIFO, 5
    printStats, 7
    remove, 7

printStats
    myFIFO, 7
printStorage
    seqSearch, 9

remove
    myFIFO, 7

search
    seqSearch, 10
seqSearch, 8
    fillStorage, 9
    printStorage, 9
    search, 10
    seqSearch, 9
    storage, 10
storage
    seqSearch, 10