

Stacks, Queues and Command Line Arguments

In other words, Project 3

Quiz

- If you have a class with dynamic data, what are the three methods you need to implement?

Stacks

- A simple FILO data structure
- Elements are added to the top and removed from the top
- How do you implement one?

Stack Implementation

- At heart, just a dynamic array.
- Ours will be of size 10 to start
- Has two main operations
 - Push
 - adds element to top of stack
 - Pop
 - removes elements from top of stack
 - Both should return a bool to indicate success

More Ideas

- Also nice to include some maintenance functions:
 - clear
 - print
 - isempty
 - isfull
- May want a private piece of data to indicate where the top is.

Queues

- A queue is a simple FIFO data structure.
- Think of waiting in line to check-out of a store.
- Not as easy to implement as it seems.
- What happens if you use a simple linear array with repeated insertions and deletions?
 - The head and tail move causing problems.

Solution

- Make the queue circular.
- The problem now becomes when is the queue empty and full?
- Solution
- Leave one cell empty.
- So in our queue the initial size will be 11.
- The trade-off is one empty cell for processing time.

States of the Queue

- Empty:
 - How do you tell
 - Rear of queue equal Front of queue
- Full:
 - How do you tell
 - Rear of queue plus 1 mod size of queue equals front of queue

Queue Implementation

- Mostly need:
 - Private data indicating rear and front
 - Size of queue
- Methods for:
 - Enque
 - Deque

Additional Ideas

- May also want to have:
- clear
- print
- isEmpty
- isFull

Command Line Arguments

- Main accepts two parameters:
 - char *Argv[] and int Argc
- Argv is an array of character pointers that contains the arguments with which the program was invoked.
- Argc is the number of arguments the program was invoked with.
- Argv[0] is always the name of the executable

Partial Example

```
int main( int argc, char* argv[ ] )
{
    if ( argc != 3 )
        cout << "Too few arguments\n";
    else
    {
        ifstream inFile( argv[ 1 ] );
        ...
    }
}
```

Group Activity

- Design a stack class interface, i.e. a class declaration for a stack class.
- Be sure to include functions for insertion, deletion, creation, copying assignment and destruction
- Write the implementation for the copy constructor or assignment operator.
- Turn in and leave; ask me for help if needed