Queue ADT

* lists as well
* inserts are done at one end (ENQUEUE: add an element)
* deletes are done at the other end (DEQUEUE: remove)

Examples: servers, printers

Basic Operations:

* enequeue: add element
* dequeue: remove an element
* front: retrieve element at the front

Array Implementation:

**Problem**: with simple arrays the contesnts of the queue must be shifted to the front with dequeue

**Solution**: keep track of front and back

const int QUEUE\_CAPACITY=128;

typedef int dataType;

class Queue{

public:

Queue();

bool empty() const;

void enqueue(const dataType &d);

dataType front() const;

void dequeue();

private:

int myFront;

int myBack;

dataType a[QUEUE\_CAPACITY];

};

Queue::Queue(){

myFront=0;

myBack=0;

}

bool Queue::empty() const{

return (myFront==myBack);

}

void Queue::enqueue(const dataType &d){

int newBack=(myBack+1)%QUEUE\_CAPACITY;

if (newBack!=myFront){

a[myBack]=d;

myBack=newBack;

}

else{

cerr<<"Queue Full";

}

}

dataType Queue::front() const{

if (!empty()){

return (a[myFront]);

}

else{

dataType garbage;

cerr<<"Queue empty";

return garbage;

}

}

void Queue::dequeue(){

if (!empty()){

myFront=(myFront+1)%QUEUE\_CAPACITY;

}

else{

cerr<<"Empty";

}

}

Circular array.

Circular linked list.