

# A queue class for characters

```
/*
C# A Beginner's Guide
By Schildt

Publisher: Osborne McGraw-Hill
ISBN: 0072133295
*/

/*
Project 5-2

A queue class for characters.
*/
using System;

class Queue {
    public char[] q; // this array holds the queue
    public int putloc, getloc; // the put and get indices

    public Queue(int size) {
        q = new char[size+1]; // allocate memory for queue
        putloc = getloc = 0;
    }

    // put a character into the queue
    public void put(char ch) {
        if(putloc==q.Length-1) {
            Console.WriteLine(" – Queue is full.");
            return;
        }

        putloc++;
        q[putloc] = ch;
    }

    // get a character from the queue
}
```

```
public char get() {
if(getloc == putloc) {
Console.WriteLine(" - Queue is empty.");
return (char) 0;
}

getloc++;
return q[getloc];
}
}

// Demonstrate the Queue class.
public class QueueDemo1 {
public static void Main() {
Queue bigQ = new Queue(100);
Queue smallQ = new Queue(4);
char ch;
int i;

Console.WriteLine("Using bigQ to store the alphabet.");
// put some numbers into bigQ
for(i=0; i < 26; i++) bigQ.put((char) ('A' + i)); // retrieve
and display elements from bigQ Console.Write("Contents of
bigQ: "); for(i=0; i < 26; i++) { ch = bigQ.get(); if(ch !=
(char) 0) Console.Write(ch); } Console.WriteLine(" ");
Console.WriteLine("Using smallQ to generate errors."); // Now,
use smallQ to generate some errors for(i=0; i < 5; i++) {
Console.Write("Attempting to store " + (char) ('Z' - i));
smallQ.put((char) ('Z' - i)); Console.WriteLine(); }
Console.WriteLine(); // more errors on smallQ
Console.Write("Contents of smallQ: "); for(i=0; i < 5; i++) {
ch = smallQ.get(); if(ch != (char) 0) Console.Write(ch); } }
}

[/csharp]
```