```
//
// CPTN278_A4_Queue_Bettle1.h
//
// Hal Bettle
// 16 January 2009
//
// Queue Class Body File.
//
// Queue implementation is array based.
// Queue uses an Internal Node.
// Error checking is left to the user.
//
#include "CPTN278_A4_Queue_Bettle1.h"

Queue::Queue()
{
    int i;

    // initialize the queue
    front = 0;
    back = 0;
    count = 0;
    for (i = 0; i < MAX_SIZE; i++)
    {
        array_of_nodes[i].type = "";
        array_of_nodes[i].fee = 0;
    }
}

bool Queue::Is_Empty(void)
{
    if ( count == 0 )
    {
        return true;
    }
    else
    {
        return false;
    }
}

bool Queue::Is_Full(void)
{
    if ( count == MAX_SIZE )
    {
        return true;
    }
    else
    {
        return false;
    }
}

void Queue::Enqueue(string category, int value)
{
    // Need to account for array wrap around
    array_of_nodes[front].type = category;
    array_of_nodes[front].fee = value;
    if ( front != MAX_SIZE - 1 )
    {
        front++;
    }
    else
    {
        front = 0;
    }
```

```
    count++;

    return;
}

int Queue::Dequeue(void)
{
    int i;
    i = array_of_nodes[back].fee;

    // Need to account for array wrap around
    if ( back != MAX_SIZE - 1 )
    {
        back++;
    }
    else
    {
        back = 0;
    }
    count--;
    return i;
}

int Queue::Front(void)
{
    if ( front == 0 )
    {
        display_item(MAX_SIZE - 1);
    }
    else
    {
        display_item(front - 1);
    }
    return array_of_nodes[front].fee;
}

int Queue::Back(void)
{
    display_item(back);
    return array_of_nodes[back].fee;
}

void Queue::display_item(int item)
{
    cout << "Item type is = "
         << array_of_nodes[item].type
         << " and fee = "
         << array_of_nodes[item].fee
         << '.'
         << endl;
    return;
}

Queue::~Queue()
{
    // Destructoe not needed.
}
```