```
class Queue {

  private int    front;              // the array index of the front item of the queue
  private int    rear;               // the array index for the next item to insert
  private int    count;              // the number of items in the queue
  private int    capacity;           // the number of available array positions
  private int    capacityIncrement;  // the amount to increase the capacity
                                     // during array expansion
  private Object[ ] itemArray;       // the array that holds queue items

/*-----------------*/

  // here, we need the no-arg constructor

  public Queue() {
      front          =  0;
      rear           =  0;
      count          =  0;
      capacity       = 10;
      capacityIncrement = 5;
      itemArray = new Object[capacity];
  }

/*-----------------*/

  public boolean empty() {
      return (count == 0);
  }

/*-----------------*/

  public void insert(Object newItem) {

   // if the itemArray does not have enough capacity,
   // expand the itemArray by the capacity increment
      if (count == capacity) {
          capacity += capacityIncrement;
          Object[ ] tempArray = new Object[capacity];
          if (front < rear) {          // if the items are in itemArray[front:rear-1]
              for (int i = front; i < rear; i++) {
                  tempArray[i] = itemArray[i];
              }
          } else {          // otherwise, move the items in two separate sections
              for (int i = 0; i < rear; i++) {                    // section one:
                  tempArray[i] = itemArray[i];                // itemArray[0:rear-1]
              }
              for (int i = front; i < count; i++) {                // and section two:
                                                          // itemArray[front:count-1]
                  tempArray[i+capacityIncrement] = itemArray[i];
              }
              front += capacityIncrement;        // then change front to point to
                                                 // its new position
          }
          itemArray = tempArray;
      }
```

If count == capacity then front == rear so the Boolean expression front < rear is false. As a result, this code will never be executed.

M. Suchenek 3/14/2011