

Instructor: Krishna Mahavadi

These problems were given on exams for this course. Some older problems did not make use of generics in Java, but generic implementations are now required in this course.

**Problem 1** The generic class Queue is to be programmed with an array based implementation. A partial version of the implementation follows. The two most important methods have been omitted. The treatment of an empty Queue (as given by the constructor) is different from the one we used in class. Make sure that your methods work correctly with the choices made by the constructor.

```
public class P2<T> implements Queue<T> {
    private T data[];
    private int front, rear, size;
    public P2() { data = (T[]) new Object[100]; front = size = 0; rear = -1; }
    public int size() { return size; }
    public boolean empty() { return size == 0; }
    // methods omitted here
}
```

- (a) **Identify the two missing methods.** For each give the name, parameters and return type.
- (b) **Give a complete implementation** of **ONE** of the two missing Queue methods. (You can choose either of them.)

**Problem 2** (i) Write the Java interface for a generic ADT Stack. (Just write an interface that specifies methods — **do not** implement the methods.)

(ii) Write the Java interface for a generic ADT Iterator. (Just write an interface that specifies methods — **do not** implement the methods.)

(iii) Write a client function with title line:

```
public static <T> int count(Stack<T> s, T x)
```

The function returns a count of the number of times the object  $x$  is found on the Stack  $s$ . When the method ends the Stack should store exactly the same data as at the beginning of the method. However your function will need to alter the Stack data as it runs.

(iv) Write a client function with title line:

```
public static <T> int count(Iterator<T> i, T x)
```

The function returns a count of the number of times the object  $x$  is found on the Iterator  $i$ . This function can and should empty out the Iterator and should not restore it to its original state.

**Problem 3** The following skeleton of a generic class Queue is missing the methods enqueue and dequeue. Give Java implementations for the methods. The queue is to be implemented with singly linked nodes. You do not need to implement the Node class, and you may assume that it has standard methods.

```
public class Queue<T> {
    private Node<T> front, rear;
    private int size;
    public Queue() {
        front = rear = null;
        size = 0;
    }
    public int size() {
        return size;
    }
}
```

```
}  
public boolean empty() {  
    return size == 0;  
}  
public T dequeue() // give an implementation for this  
public void enqueue(T x) // give an implementation for this  
}
```