CMSC 341 Data Structures
Stack and Queue Review

October 26, 2010

These are some review questions on stacks. The class definitions for stack and queue are provide at the end of the questions.

Stacks

1. Using only the operations of the stack, write a function that determines if a string is a palindrome (i.e. reads the same backward and forward; e.g. "level"). The prototype for this function is given below.

```
bool IsPalindrome(String theString);
```

2. What is the output of the following code?

```
int [] values = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19 };
Stack<Integer> s;

for (int 1 = 0; i < values.length; i++)
     s.push( values[ i ] );

int n = 25;
for (int i = 0; i < 4; i++)

{
     n += s.top( ); s.pop( );
}
 for (int i = 0; i < 2; i++)
 {
     n -= s.top( ); s.pop( );
 }
 System.out.println( n );
```

1. Discuss the advantages and disadvantages of the text's array implementation and the lecture notes layered implementation of the stack ADT. At a minimum, consider the asymptotic time performance of the isEmpty( ), pop( ) and push( ) operations.

2. Using only the operations of the stack given in the class definition (i.e. without using the stack's copy constructor or assignment operator), write a Java function that returns a copy of the user specified stack. The prototype for the function is given below

```
Stack<AnyType> CopyStack(Stack<AnyType> otherStack)
```

Queues

1. Using the operations of the stack and queue, write a function that determines if a string is a palindrome (i.e. reads the same backward and forward; e.g. "level"). The prototype for this function is given below.

```
bool IsPalindrome(String theString );
```

2. Suppose that Q is an initially empty array-based queue of size 5. Show the values of the data members front and back after each statement has been executed. Indicate and errors that might occur.

```
Queue<Character> Q( 5 );          front = _____   back = _____
Q.enqueue( 'A' );               front = _____   back = _____
Q.enqueue( 'B' );               front = _____   back = _____
Q.enqueue( 'C' );               front = _____   back = _____
char c = Q.dequeue( );          front = _____   back = _____
Q.enqueue( 'A' );               front = _____   back = _____
```

3. Discuss the advantages and disadvantages of the linked list and array-based implementations of a queue.

4. Describe three "real life" applications of a queue.

5. Explain how to implement a queue using two stacks.

Definition of the Stack Class
This is the definition of the array based stack from the text .

```
public class Stack<AnyType>
{
    public Stack( int capacity );
    public boolean isEmpty( );
    public boolean isFull ( );
    public AnyType top( );
    public void makeEmpty( );
    public void AnyType pop( );
    public void push( AnyType element );
    public AnyType topAndPop( ) ;
    private ArrayList< AnyType > theArray;
    private int topOfStack;
}
```

Definition of the Queue Class
This is the definition of the array based queue from the text.

```
public class Queue
{
    public Queue( int capacity );
    public boolean isEmpty( );
    public boolean isFull ( );
    public AnyType getFront( );
    public void makeEmpty( );
    public AnyType dequeue( );
    public void enqueue( T element );
    private ArrayList< AnyType > theArray;
    private int currentSize, front, back;
    private void increment( int x );
};
```