

## Data Structures

### Prof. Loftin: Practice Test Problems for Test 2

1. Assume that `Stack` and `Queue` are implementations of the corresponding ADTs using Java generics. Consider the following method

```
public static void problem1(Stack<E> s){
    Queue<E> q = new Queue<E>();
    while (! s.empty())
        q.enqueue(s.pop());
    while (! q.empty())
        s.push(q.dequeue());
}
```

- (a) Let `stack` be a stack of `Integers` containing the data

7  
10  
5  
-2

How is `stack` changed (if at all) when `problem1(stack)` is called?  
Trace through the method carefully.

- (b) If `st` is any stack, what is the effect of calling `problem1(st)`?  
Justify your answer.

2. Trace through the state of the stack `s` in the following code fragment.

```
Stack<String> s = new Stack<String>();
s.push("happy");
s.push("sad");
String st = s.peek();
s.push("numb");
s.push(st+"dle");
s.pop();
st = s.pop();
s.push(st);
```

3. Trace through the state of the queue `q` in the following code fragment. (Assume `Queue` is an implementation of the standard queue interface using Java generics.)

```
Queue<Integer> q = new Queue<Integer>();
q.enqueue(5);
q.enqueue(7);
q.enqueue(13);
q.dequeue();
Integer t = q.peek();
q.enqueue(12+t);
q.dequeue();
q.enqueue(q.dequeue());
```

4. Evaluate the following postfix expression

3 5 7 2 8 + - \* + 4 -

Show your work.

5. Fully parenthesize the following Java expression (using the standard Java rules of precedence of operations and left-right associativity).

`x + 3 / ( y * 2 - 4 ) * w - 1`

6. What is the output to the screen?

```
public class Problem6{
    public static void main (String[] args){
        try{
            for (int i=1; i<=10; i++)
                System.out.println(g(i));
        }
        catch (Exception e){
            System.out.println("Exception caught.");
        }
        int[] ar = new int[10];
        for (int i=0; i<10; i++)
            ar[i] = i*i;
        ar[10] = 500;
        System.out.println("End of main.");
    }
    public static int g(int k){
        return 120/(7-k);
    }
}
```

7. Consider the language whose sentences are given by  $\langle W \rangle$

$\langle W \rangle = \langle W \rangle t \mid t \mid \langle W \rangle \langle S \rangle$   
 $\langle S \rangle = a \mid b \mid c$

Write a Java method

```
public static boolean inW (String s)
```

which determines whether the string  $s$  is in the language of  $W$ . (Recall that the method call `s.substring(i,j)` returns the substring of  $s$  going from index  $i$  to index  $j-1$  inclusive.)

8. Consider an implementation `StackReferenceBased` which implements the `StackInterface`, uses Java generics, and throws a `StackException`. The implementation uses a linked list structure with `top` as the head of the list.

Write the code for the method

```
public E pop() throws StackException
```

The class `StackException` is given by

```
public class StackException extends RuntimeException{  
    public StackException (String s){  
        super(s);  
    }  
}
```