1. Suppose you need to create a Stack of ints and you decide to do it with a linked structure.
   a) Draw a picture of your stack after you push data elements 1 and then 3. Include in your picture any labels you refer to in your code.

b) Give code for methods
   ```java
   void push(int x)
   int pop()
   ```
   You can make any assumptions you want about the Node class for your structure. The pop method should throw an EmptyStackException if you try to pop an empty stack.
2. Suppose class MyList holds a list of integers using a linked implementation similar to the way you implemented Queues in Lab 3. Here is a version of SelectionSort for MyLists

```java
public static void SelectionSort( MyList L ) {
    int n = L.size();
    for (int i = 0; i < n-1; i++) {
        int small = i;
        for (int j = i; j < n; j++) {
            if (L.get(j) < L.get(small))
                small = j;
        }
        int temp = L.get(small);
        L.set(small, L.get(i));
        L.set(i, temp);
    }
}
```

Give a Big-O analysis of the running time of this function on a list of size n. You only need the answer if you are right, but a sentence of explanation might help get you some partial credit if you are wrong.
3. Consider the following abstract class that holds two objects of class E:

```java
public abstract class Pair<E> {
    abstract E getFirst(); // return the first element of the pair
    abstract E getSecond(); // returns the second element
    abstract E setFirst(E item);
    abstract E setSecond(E item);
    void isTwin() {
        return getFirst() == getSecond();
    }
    void switcheroo() {
        E temp = getFirst();
        setFirst(setSecond());
        setSecond(temp);
    }
}
```

Explain what is involved in making a non-abstract version of Pair<E>. You don’t need to write any code; just say what needs to be done.
4. I am implementing an ArrayList class as you did in Lab 2. My class starts:

```java
public class MyArrayList<E> extends AbstractList<E> {
    protected int size;
    protected E[] data;

    public MyArrayList(int startSize) {
        size = 0;
        data = (E[]) new Object[startSize];
    }

    ..... Write code for a method trim() that reduces the data array so that the list completely fills the array. If the list has 5 elements trim() should reduce the array to length 5.
```
5. Integer array Nums contains positive numbers.
   
a. Write a recursive function boolean SumTo(int target) that returns true if target can be written as a sum of entries from Nums. For example, if Nums is the array {21, 13, 6} we can write 39 as 13+13+13, 40 as 21+13+6 and 42 as 21+21, but we can't write 41 as a sum of those numbers, so SumTo(39), SumTo(40) and SumTo(42) all return true and SumTo(41) returns false.

b. Describe in English how you would apply the Dynamic Programming technique to SumTo(). You will probably want to save results in an array; what is the type of this array and how should its entries be initialized? Note that you can answer part (b) even if you couldn’t answer part(a).
This is extra space. If you want me to grade anything here please indicate that clearly.

Please write and sign the Honor Pledge when you are finished with the exam.