Exercise 1

```
65    public int remove(int idx) {
66        int value = 0;
67        if (front == null) {
68            throw new IndexOutOfBoundsException();
69        }
70        if (idx == 0) {
71            value = front.data;
72            front = front.next;
73        } else {
74            ListNode current = front;
75            int i = 0;
76            while (current.next != null && i < idx-1) {
77                current = current.next;
78                i++;
79            }
80            if (current.next == null) {
81                throw new IndexOutOfBoundsException();
82            }
83            value = current.next.data;
84            current.next = current.next.next;
85        }
86        size--;
87        return value;
88    }
```

Diamond1 → 1 of 2 branches missed
Diamond 2 → 1 of 2 branches missed
Diamond 3 → 3 of 4 branches missed
Diamond 4 → 1 of 2 branches missed

Exercise 2

```
    public static void add(ListNode list, int data){
        while (list != null) {
            list = list.next;
        }

        list.next = new ListNode(data);
    }
```

(Two overlapping notifications)

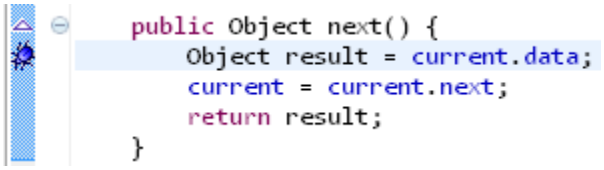
1 - Null pointer dereference of list in edu.ncsu.csc216.linked_list.ListNode.add(ListNode, int)

A null pointer is dereferenced here. This will lead to a NullPointerException when the code is executed.

2 - Load of known null value in edu.ncsu.csc216.linked_list.ListNode.add(ListNode, int)

The variable referenced at this point is known to be null due to an earlier check against null. Although this is valid, it might be a mistake (perhaps you intended to refer to a different variable, or perhaps the earlier check to see if the variable is null should have been a check to see if it was non null).

Exercise 3

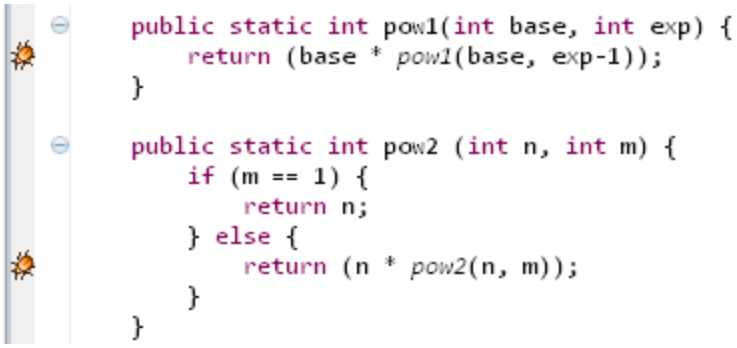


```
public Object next() {
    Object result = current.data;
    current = current.next;
    return result;
}
```

linked_List_Iterator_Bug.LinkedIterator.next() can't throw NoSuchElementException

This class implements the java.util.Iterator interface. However, its next() method is not capable of throwing java.util.NoSuchElementException. The next() method should be changed so it throws NoSuchElementException if is called when there are no more elements to return.

Exercise 4



```
public static int pow1(int base, int exp) {
    return (base * pow1(base, exp-1));
}

public static int pow2 (int n, int m) {
    if (m == 1) {
        return n;
    } else {
        return (n * pow2(n, m));
    }
}
```

Bug: There is an apparent infinite recursive loop in infinite_Recursion_Bug.Recursive_Methods.pow1(int, int)

This method unconditionally invokes itself. This would seem to indicate an infinite recursive loop that will result in a stack overflow.