

# CSE 143: Computer Programming II

---

## Lecture 6 Exercises Solutions

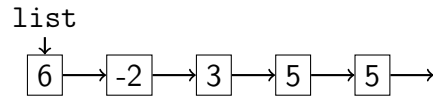
(a) What Does `mystery()` print?

```
1  public static void mystery() {
2      List<Integer> list1 = new ArrayList<Integer>();
3      list1.add(8);
4      list1.add(3);
5
6      List<Integer> list2 = new ArrayList<Integer>();
7
8      list2.add(100);
9
10     List<Integer> list3 = list2;
11
12     list2 = list1;
13
14     list2.add(5);
15
16     list1.add(2);
17
18     System.out.println("A: " + list1);
19     System.out.println("B: " + list2);
20     System.out.println("C: " + list3);
21 }
```

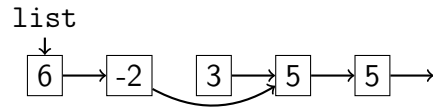
*Solution:* See slides.

### (b) List-To-List

Given the following linked list:



Write code to result in the following list:

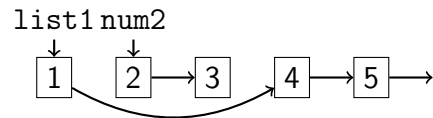


*Solution:*

```
1 list.next.next = list.next.next.next;
```

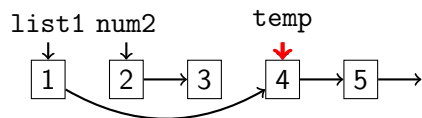
### (c) List-To-Code

Given the following linked list and variables `list1` and `num2`, write code to fix the list so it's in numerical order:

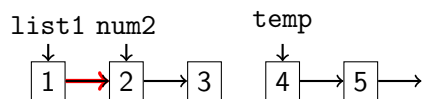


*Solution:*

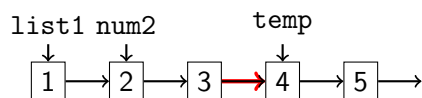
```
1 ListNode temp = list1.next;
```



```
2 list.next = num2;
```



```
3 num2.next.next = temp;
```



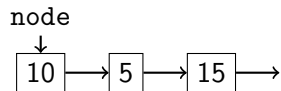
(d) Together

Consider the following code. Draw the resulting linked list.

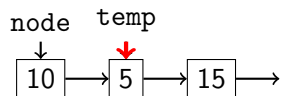
```
1 ListNode node = new ListNode(10, new ListNode(5, new ListNode(15)));
2 ListNode temp = node.next;
3 temp.data = 100;
4 node.next = new ListNode(12, node.next.next.next);
5 node.next.next = temp;
```

*Solution:*

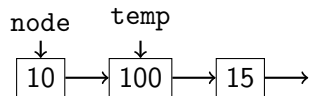
```
1 ListNode node = new ListNode(10, new ListNode(5, new ListNode(15)));
```



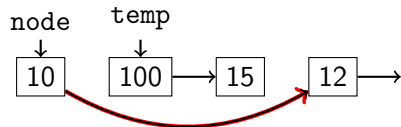
```
2 ListNode temp = node.next;
```



```
3 temp.data = 100;
```



```
4 node.next = new ListNode(12, node.next.next.next);
```



```
5 node.next.next = temp;
```

