1) O(n), must traverse whole list for a given int index.

2) O(n), If have to find node first, if not, O(1)

3) Add at head, O(1)
Add at end O(n), overhead retrieve node
Add at index O(n)

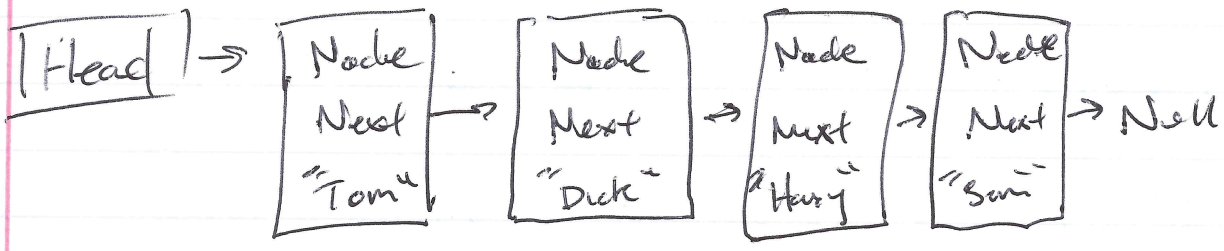
4) Head → 5 → 10 → 7 → 30 → null

```
int sum=0;
Node<Integer> nodeRef = head;
while(nodeRef != null) {
    int next = nodeRef.data;
    sum += next
    nodeRef = nodeRef.next;
}
```

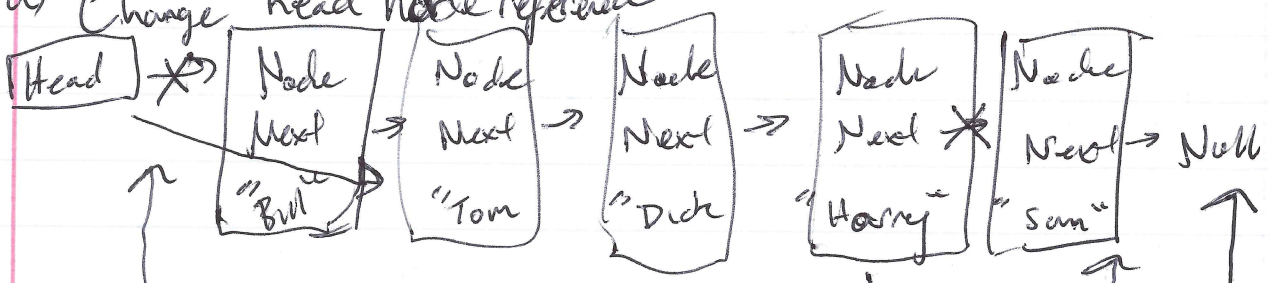
5) a. Replaces head node with Shakira.
b. Cuts out the second person in the list
c. Adds Tanika to the end
d. Begins at head, goes until it hits null or Harry, ~~take the node off~~ if the node is Harry, change Harry to Sally and add Harry after Sally.
Essentially: if Harry is on the list, add Sally before him

Zach Chero

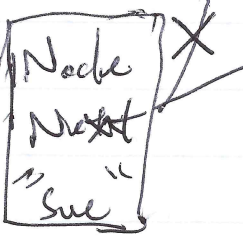
Programming Exercise 2 Section 2.5



a) Change head node reference



b) Insert Sue as Harry's new next and link Sue to Harry's last child, Sam.



c) To remove Bill we simply change the head node and Bill will be collected by garbage. Then we redefine the head reference

d) Remove "Sam"

To remove Sam we simply stop referencing him from Sue.