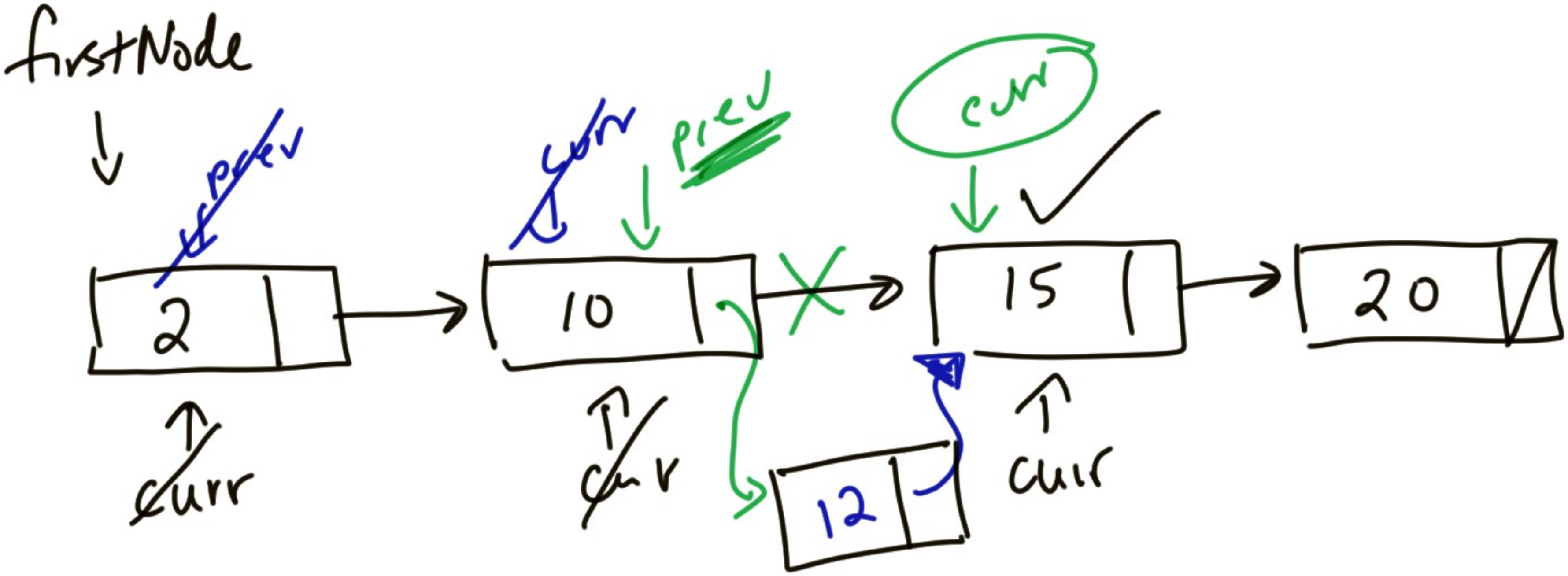
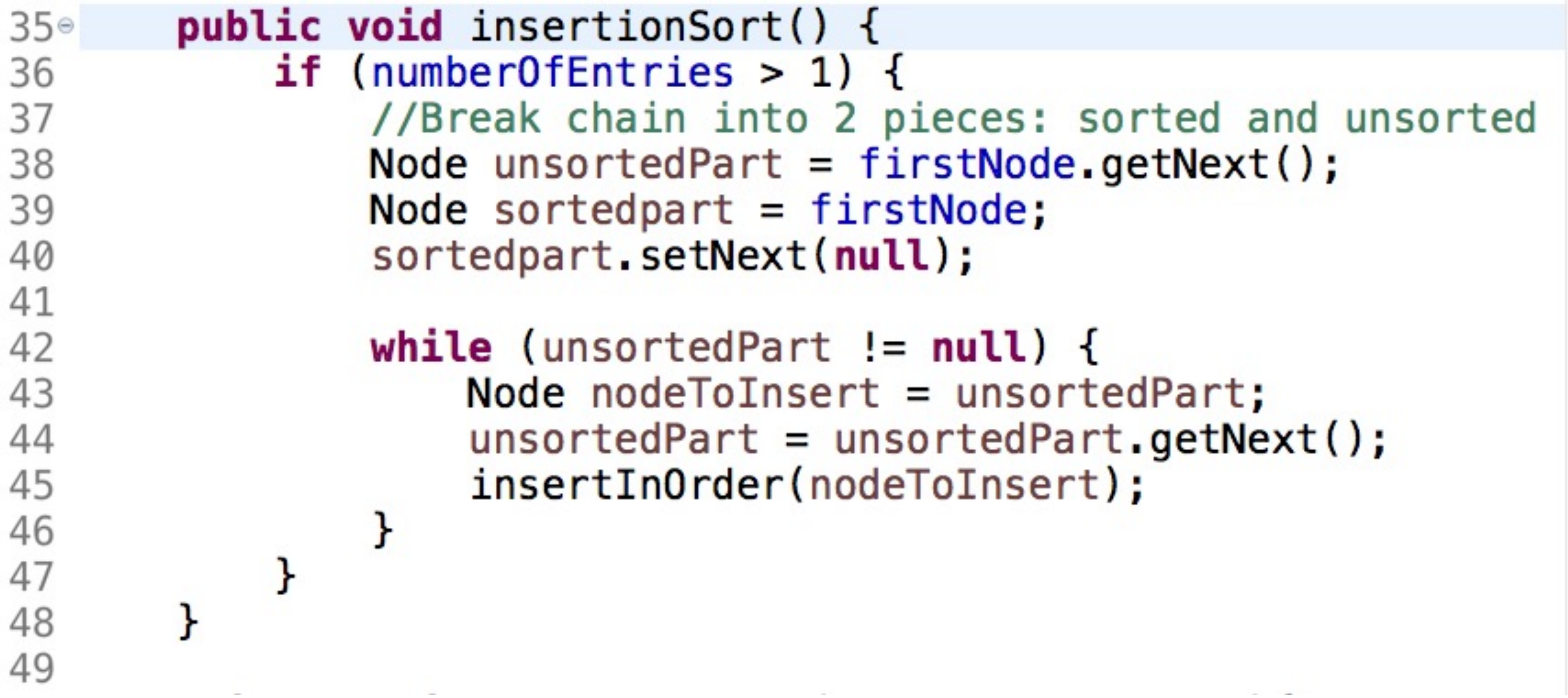


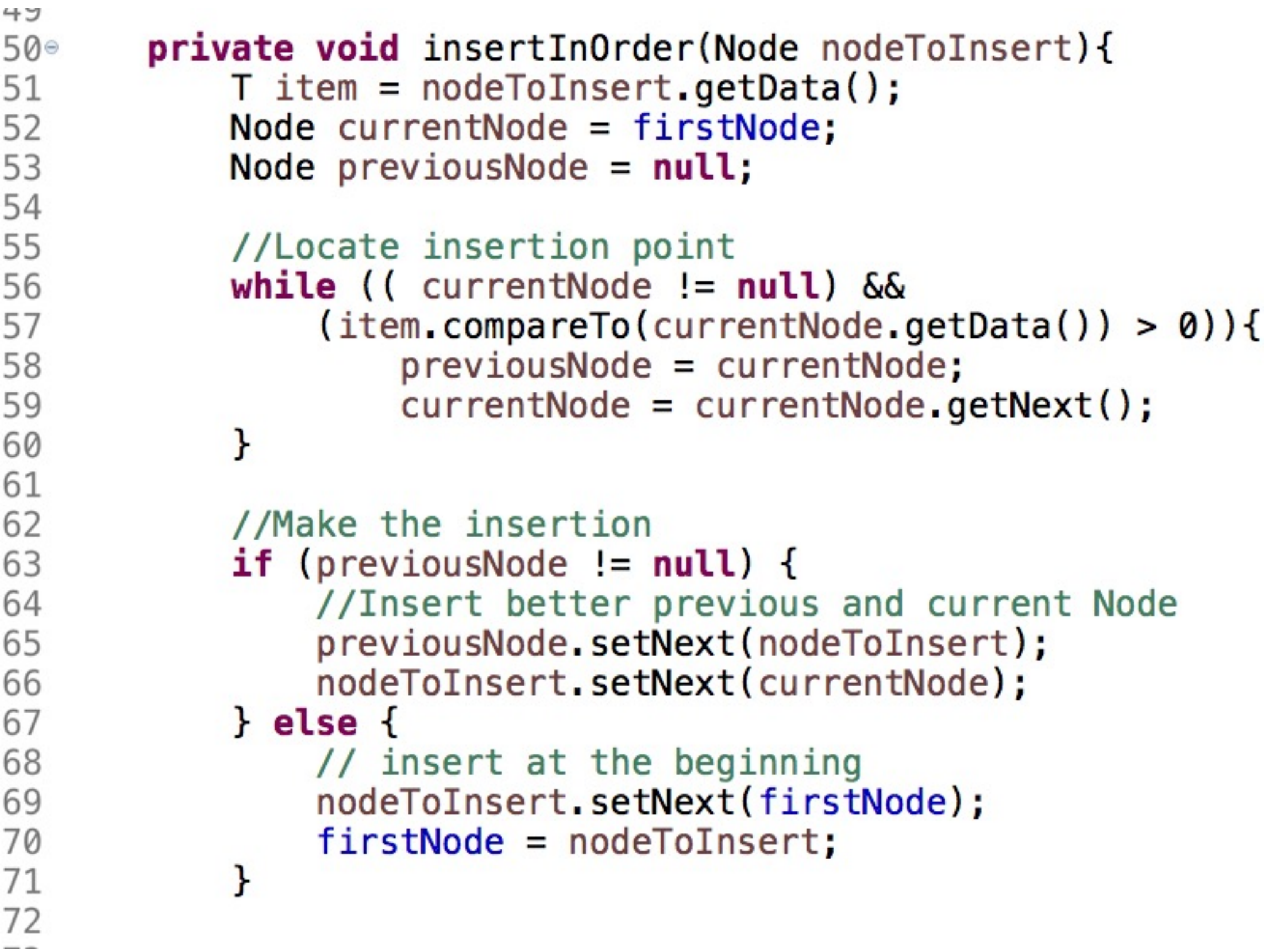
1) figure out where 12 belongs ✓

2) newnode inserted correctly

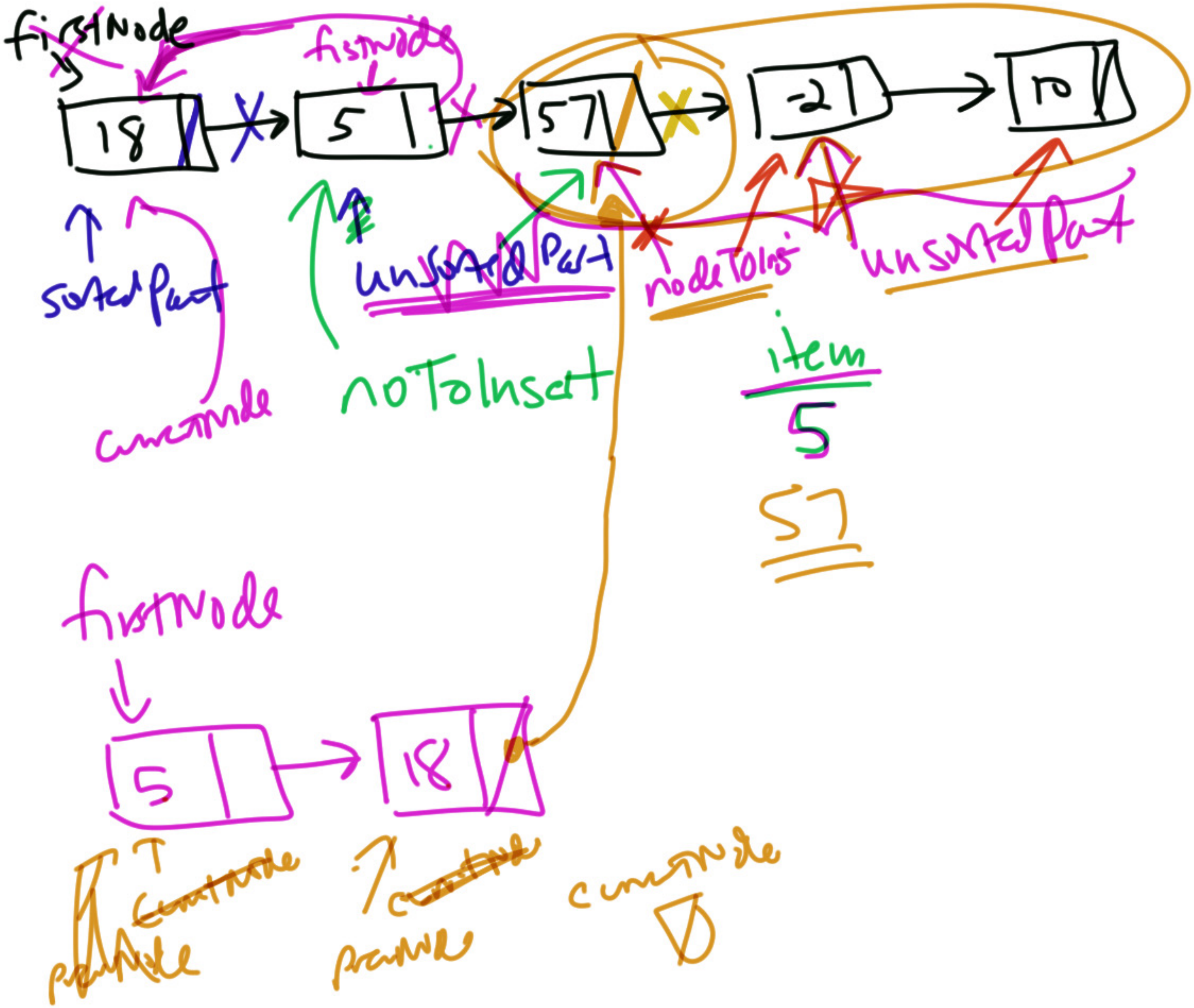
Insert into an ordered list



```
    public void insertionSort() {
        if (numberOfEntries > 1) {
            //Break chain into 2 pieces: sorted and unsorted
            Node unsortedPart = firstNode.getNext();
            Node sortedpart = firstNode;
            sortedpart.setNext(null);

            while (unsortedPart != null) {
                Node nodeToInsert = unsortedPart;
                unsortedPart = unsortedPart.getNext();
                insertInOrder(nodeToInsert);
            }
        }
    }
```



```
    private void insertInOrder(Node nodeToInsert){
        T item = nodeToInsert.getData();
        Node currentNode = firstNode;
        Node previousNode = null;

        //Locate insertion point
        while (( currentNode != null) &&
            (item.compareTo(currentNode.getData()) > 0)){
                previousNode = currentNode;
                currentNode = currentNode.getNext();
        }

        //Make the insertion
        if (previousNode != null) {
            //Insert better previous and current Node
            previousNode.setNext(nodeToInsert);
            nodeToInsert.setNext(currentNode);
        } else {
            // insert at the beginning
            nodeToInsert.setNext(firstNode);
            firstNode = nodeToInsert;
        }
```


firstNode
firstNode
18
5
57
-2
10
sortedPart
unsortedPart
currNode
noToInsert
nodeToIns
unsortedPart
item
5
57
firstNode
5
18
~~currNode~~
prevNode
~~currNode~~
prevNode
currNode
Ø