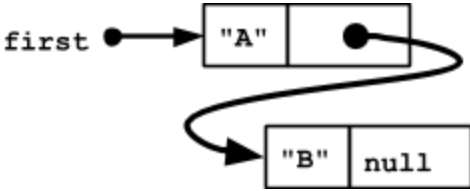
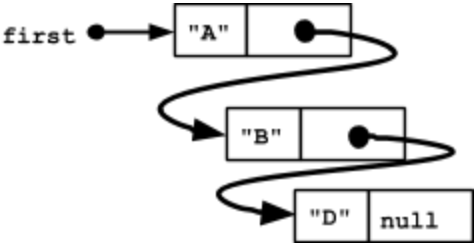
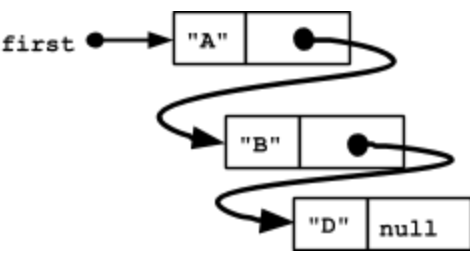
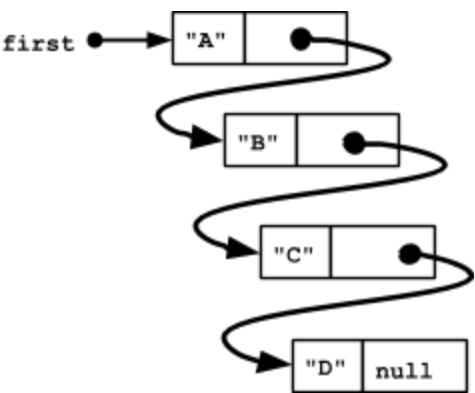


# LINKED LISTS WORKSHEET

3. Insert a new Node with item = "D" at the end of the linked list from (2)

Before	Java code	After
	<pre>Node temp = new Node("D"); first.next.next = temp;</pre>	

4. Insert a new Node with item = "C" after the Node with item = "B" in the list from (3)

Before	Java code	After
	<pre>Node temp = new Node("C"); temp.next = first.next.next; first.next.next = temp;</pre>	

5. Using a do-while loop, print all the items in the linked list from (4).

```
if (first == null)
    StdOut.println("EMPTY LIST");
else {
    Node current = first;
    do {
        StdOut.println(current.item);
        current = current.next;
    } while (current != null);
}
```

6. Suppose you have a linked list. The start of the list is stored in Node first. Write a loop to add a new Node with item = "E" to the end of linked list. The list may contain 0 or more items.

```
Node temp = new Node("E");
if (first == null)
    first = temp;
else {
    Node current = first;
    while (current.next != null)
        current = current.next;
    current.next = temp;
}
```