```
def delete_at_given_position(self, target_position):
        # case 1: List is empty
        if (self.head  == None):
            print("List is empty")
            return

        # case 2: position is invalid

        if (target_position <= 0):
            print(f"Invalid target Position: {target_position}")
            return

        # case 3: single node
        if ( self.head.next == None):
            self.head = None
            return

        # case 4: multi node
        to_be_deleted = self.head
        current_position = 1

        while(current_position < target_position and to_be_deleted is not None):
            current_position = current_position + 1
            to_be_deleted = to_be_deleted.next

        if to_be_deleted == None:
            print(f" Target position {target_position} is invalid, we have lesser nuber of node. ")
            return

        # case 4.1:  to_be_deleted node is the last node
        if (to_be_deleted.next == None):
            to_be_deleted.prev.next = None
```

```
    return

# case 4.2: there are nodes after the to_be_deleted Node
to_be_deleted.next.prev = to_be_deleted.prev
to_be_deleted.prev.next = to_be_deleted.next
```