LINKED LIST IMPLEMENTATION OF STACK

```
class Node:
    def __init__(self, book_title):
        self.book_title = book_title
        self.next = None

class Stack:
    def __init__(self):
        self.top = None

    def push(self, book_title):
        """Add a book title to the top of the stack."""
        new_node = Node(book_title)
        new_node.next = self.top
        self.top = new_node
        print(f"Book '{book_title}' added to the stack.")

    def pop(self):
        """Remove and return the top book title from the stack."""
        if self.is_empty():
            print("Stack is empty. No book to pop.")
            return None

        popped_node = self.top
        self.top = self.top.next
        popped_title = popped_node.book_title
        del popped_node
        print(f"Book '{popped_title}' removed from the stack.")
        return popped_title

    def peek(self):
        """Return the top book title without removing it."""
        if self.is_empty():
            print("Stack is empty. No book to peek.")
            return None
        return self.top.book_title

    def is_empty(self):
        """Check if the stack is empty."""
        return self.top is None

    def display_stack(self):
        """Display all book titles in the stack."""
        if self.is_empty():
            print("Stack is empty.")
            return
```

```
        current = self.top
        print("Current Stack of Books:")
        while current:
            print(f"- {current.book_title}")
            current = current.next

# Example Usage
if __name__ == "__main__":
    book_stack = Stack()

    # Adding books to the stack
    book_stack.push("The Catcher in the Rye")
    book_stack.push("To Kill a Mockingbird")
    book_stack.push("1984")

    # Displaying the stack
    book_stack.display_stack()

    # Peeking at the top book
    top_book = book_stack.peek()
    if top_book:
        print(f"Top book on the stack: '{top_book}'")

    # Removing books from the stack
    book_stack.pop()
    book_stack.pop()

    # Displaying the stack after pops
    book_stack.display_stack()

    # Attempting to pop from an empty stack
    book_stack.pop()  # Should indicate stack is empty
    book_stack.pop()  # Should also indicate stack is empty
```