class Node:

"""A class representing a node in a linked list."""

def \_\_init\_\_(self, data):

self.data = data

self.next = None

class Stack:

"""A class representing a stack using a linked list."""

def \_\_init\_\_(self):

self.top = None

def is\_empty(self):

"""Check if the stack is empty."""

return self.top is None

def push(self, data):

"""Push an element onto the stack."""

new\_node = Node(data)

new\_node.next = self.top

self.top = new\_node

print(f"Pushed {data} onto the stack.")

def pop(self):

"""Pop an element from the stack."""

if self.is\_empty():

print("Stack is empty. Cannot pop.")

return None

popped\_node = self.top

self.top = self.top.next

print(f"Popped {popped\_node.data} from the stack.")

return popped\_node.data

def peek(self):

"""Peek at the top element of the stack without removing it."""

if self.is\_empty():

print("Stack is empty. Nothing to peek.")

return None

return self.top.data

def display(self):

"""Display all elements in the stack."""

if self.is\_empty():

print("Stack is empty.")

return

current = self.top

print("Stack elements:")

while current:

print(current.data, end=" -> ")

current = current.next

print("None")

# Example usage:

stack = Stack()

stack.push(10)

stack.push(20)

stack.push(30)

stack.display()

stack.pop()

stack.display()

print(f"Top element is {stack.peek()}")

**output**

**Pushed 10 onto the stack.**

**Pushed 20 onto the stack.**

**Pushed 30 onto the stack.**

**Stack elements:**

**30 -> 20 -> 10 -> None**

**Popped 30 from the stack.**

**Stack elements:**

**20 -> 10 -> None**

**Top element is 20**

**=== Code Execution Successful ===**