

CP411107 Data Structures and Algorithms  
การสร้าง stack ด้วย Link List

- การสร้าง Stacks ด้วย Link List หมายถึง การแทนที่ข้อมูลของ stack ด้วย Link list ซึ่งเป็นการจัดสรรเนื้อที่หน่วยความจำแบบ Dynamic นั่นคือ หน่วยความจำจะถูกจัดสรรเมื่อมีการขอใช้จริงๆ ระหว่างการประมวลผลโปรแกรมผ่านด้วย Pointer

8/23

CP411107 Data Structures and Algorithms  
การแทนที่โครงสร้างข้อมูลลงชื่อนี้ด้วยรายการแบบ

- เป็นการแทนที่ที่มีความยืดหยุ่นมาก
- เนื่องจากไม่ต้องกำหนดหรือจดหน่วยความจำล่วงหน้า
- ข้อมูลที่เก็บในกองข้อมูลจำนวนเท่าใดก็ได้ การปฏิบัติการมี 4 ขั้นตอนคือ การสร้าง, การ Push, การ Pop และการแสดง

8/23

CP411107 Data Structures and Algorithms  
การสร้างกล่องข้อมูลแบบรายการแบบ

- ยกตัวอย่างกรณีที่เป็นรายการแบบเดียว การสร้างจะเป็นดังนี้

```
typedef struct node
{
    int data;
    struct node *Next;
} node;
node *Top;
int info;
```

node

data | Next

8/23

CP411107 Data Structures and Algorithms  
การสร้างกล่องข้อมูลแบบรายการแบบ

- ยกตัวอย่างกรณีที่เป็นรายการแบบเดียว การสร้างจะเป็นดังนี้

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

node

data | Next

8/23

CP411107 Data Structures and Algorithms  
การนำข้อมูลเข้าไปในกองข้อมูลแบบรายการแบบ

- Top = Null
- New(P)
- Data(p) = 80
- Next(p) = Null

```
Top = Node(None)
P=Node(80)
P.next=None
Top =P
```

P

Top

90

80

Top = Node(None)  
P=Node(80)  
P.next=None  
Top =P

8/23

CP411107 Data Structures and Algorithms  
การนำข้อมูลเข้าไปในกองข้อมูลแบบรายการแบบ

- เพิ่มข้อมูล 90 เข้าไปในรายการแบบ

- New(P)
- Data(P) = 90
- Next(P) = Top
- Top = p

```
P=Node(90)
P.Next =Top
Top =P
```

P

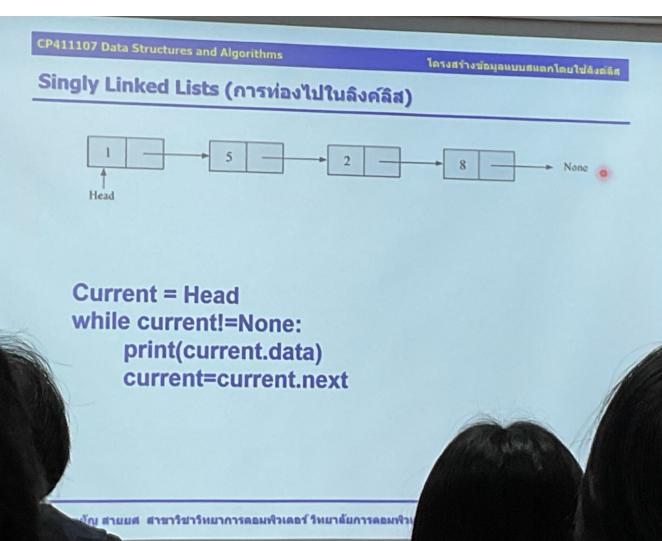
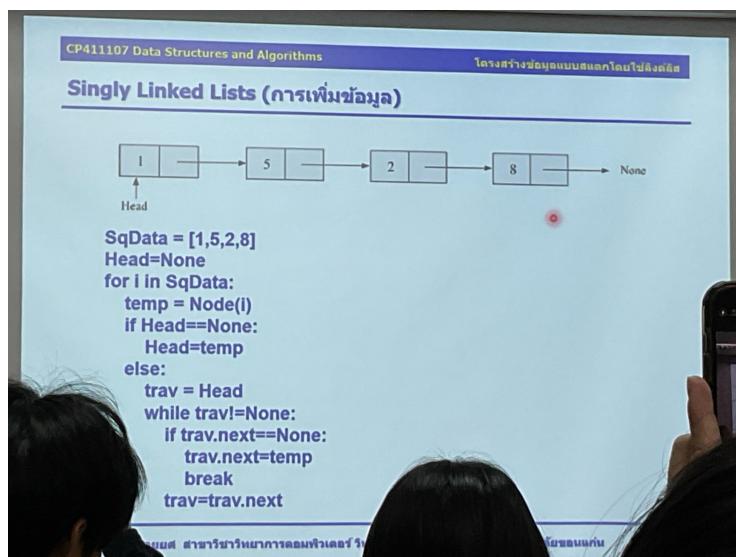
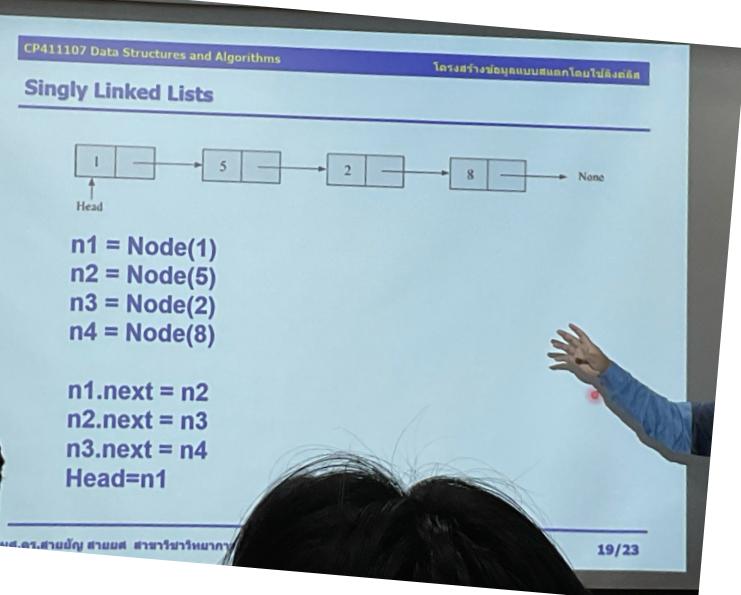
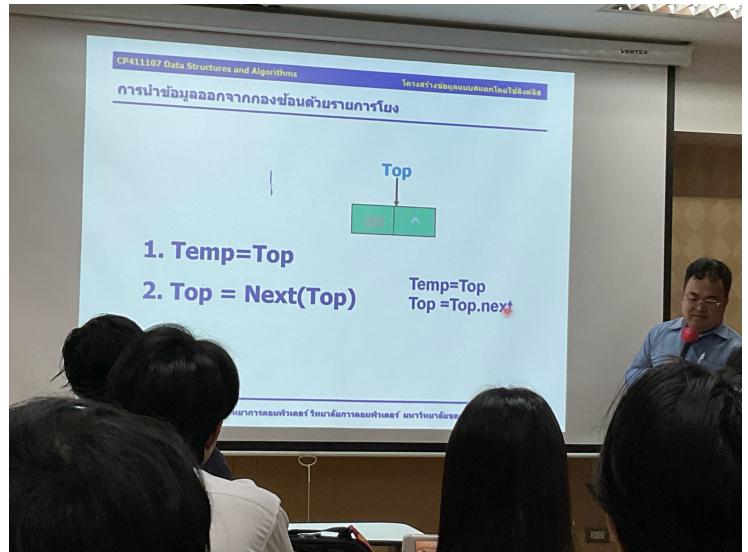
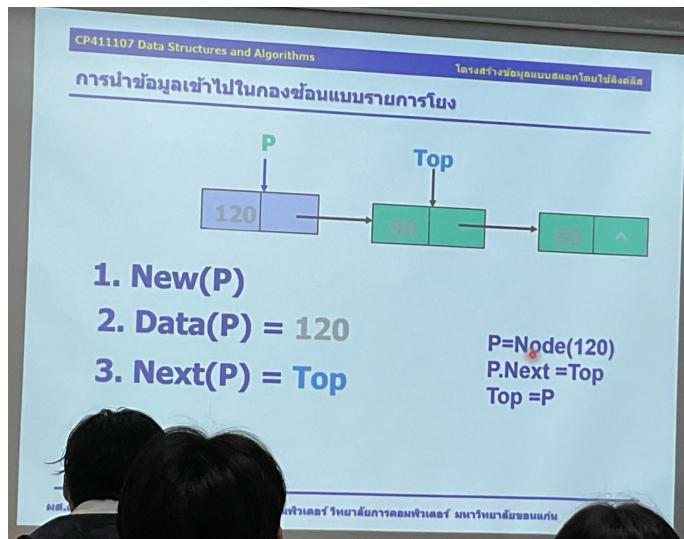
Top

90

80

Top = Node(90)  
P.Next =Top  
Top =P

8/23



```
SinglyLinklist_tail.py - C:\Users\Admin\OneDrive\Desktop\python\CP411107 Data Structure\1
File Edit Format Run Options Window Help
class Node:
    def __init__(self, data=None):
        self.data = data
        self.next = None

SqData = [1,5,2,8]
Head=None
Tail=None

for i in SqData:
    temp = Node(i)
    if Head==None:
        Head=temp
        Tail=Head
    else:
        Tail.next=temp
        Tail=Tail.next

current=Head
while current!=None:
    print(current.data)
    current=current.next
```

A screenshot of a computer monitor displaying Python code for singly linked lists. The code defines a Node class with an \_\_init\_\_ method, initializes a list SqData, creates a Head node, and then iterates through SqData to build the linked list. It prints the data of each node.

```
SinglyLinklist_for.py - C:\Users\Admin\OneDrive\Desktop\รามคำแหง\CP411107 Data Structure\1_f\WS\code\SinglyLinklist_for.py
File Edit Format Run Options Window Help
def __init__(self, data=None):
    self.data = data
    self.next = None

it("++SqData = [1,5,2,8]
Head=None
ext=n
for i in SqData:
    temp = Node(i)
    if Head==None:
        Head=temp
    else:
        trav = Head
        while trav!=None:
            if trav.next==None:
                trav.next=temp
= n1
e tra
print(trav= current=Head
while current!=None:
= n1
    print(current.data)
    current=current.next
```

