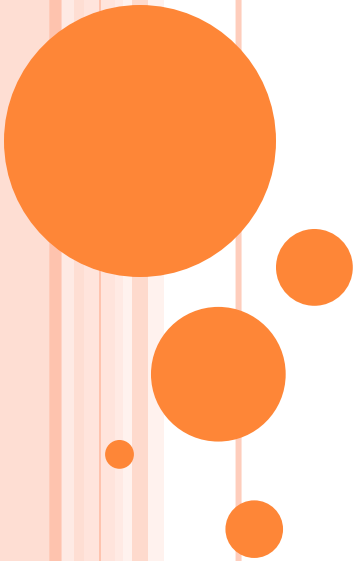

Data Structure - II

Stack and Queues

STACK

- A stack is a collection of data items that can be accessed at only one end, called top.
- Items can be inserted and deleted in a stack only at the top.
- The last item inserted in a stack is the first one to be deleted.
- Therefore, a stack is called a Last-In-First-Out (LIFO) data structure.
- 2 mains operations on Stack is PUSH & POP
- PUSH means inserting new item at top and POP means deleting item from top.

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Other Stack Term

- Peek : getting the most recent value of stack i.e value at TOP
- OverFlow : a situation when we are Pushing item in Stack that is full.
- Underflow : a situation when we are Popping item from empty stack

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

IMPLEMENTING STACK IN PYTHON

```
def isEmpty(S):
        if len(S)==0:
                return True
        else:
                return False
def Push(S,item):
        S.append(item)
        top=len(S)-1

def Pop(S):
        if isEmpty(S):
                return "Underflow"
        else:
                val = S.pop()
                if len(S)==0:
                        top=None
                else:
                        top=len(S)-1
                return val
```

This function will check Stack is empty or not

This function will add new item in Stack, here setting top is mandatory

This function is used to remove item from stack, also perform checks before deletion

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

IMPLEMENTING STACK IN PYTHON

```
def Peek(S):
        if isEmpty(S):
                return "Underflow"
        else:
                top=len(S)-1
                return S[top]

def Show(S):
        if isEmpty(S):
                print("Sorry No items in Stack ")
        else:
                t = len(S)-1
                print("(Top)",end=' ')
                while(t>=0):
                        print(S[t],"<==",end=' ')
                        t-=1
                print()
```

This function will return the top most item from the stack

This function will display stack items

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

IMPLEMENTING STACK IN PYTHON

```
# main begins here
S=[]                #Stack
top=None
while True:
        print("**** STACK DEMONSTRATION ******")
        print("1. PUSH ")
        print("2. POP")
        print("3. PEEK")
        print("4. SHOW STACK ")
        print("0. EXIT")
        ch = int(input("Enter your choice :"))
        if ch==1:
                val = int(input("Enter Item to Push :"))
                Push(S,val)
```

Displaying menu to user to interact

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

IMPLEMENTING STACK IN PYTHON

```
elif ch==2:
        val = Pop(S)
        if val=="Underflow":
                print("Stack is Empty")
        else:
                print("\nDeleted Item was :",val)
elif ch==3:
        val = Peek(S)
        if val=="Underflow":
                print("Stack Empty")
        else:
                print("Top Item :",val)
elif ch==4:
        Show(S)
elif ch==0:
        print("Bye")
        break
```

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

QUEUE

- Queue is a linear list which follows FIFO approach.
- Queue allows addition of element only at one end called REAR (end of list) & Deletion of element only from FRONT end (beginning of list).
- The operation of addition and deletion is known as Enqueue and Dequeue respectively.
- Applications of Queue:
  - Printer Spooling
  - CPU Scheduling
  - Mail Service
  - Keyboard Buffering
  - Elevator

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

Implementing Queue in Python :: Queue Operations ::

- Peek : getting first value of QUEUE i.e. of FRONT position.

  e.g. **Queue[Front]** ***# Front is an int storing index of first element of queue***

- Enqueue: addition of new item in QUEUE at REAR position.

  e.g. **Queue.append(Item)**

- Dequeue: removal of item from the beginning of QUEUE.

e.g. **Queue.pop(0)**

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

PYTHON CODE: QUEUE

```
def isEmpty(Q):
        if len(Q)==0:
                return True
        else:
                return False

def Enqueue(Q,item):
        Q.append(item)
        if len(Q)==1:
                front=rear=0
        else:
                rear=len(Q)-1

def Dequeue(Q):
        if isEmpty(Q):
                return "Underflow"
        else:
                val = Q.pop(0)
        if len(Q)==0:
                front=rear=None
        return val
```

This function will check Queue is empty or not

This function will add new item in Queue, here setting top is mandatory

This function is used to remove item from Queue, also perform checks before deletion

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

PYTHON CODE: QUEUE

```
def Peek(Q):
        if isEmpty(Q):
                return "Underflow"
        else:
                front=0
                return Q[front]

def Show(Q):
        if isEmpty(Q):
                print("Sorry No items in Queue ")
        else:
                t = len(Q)-1
                print("(Front)",end=' ')
                front = 0
                i=front
                rear = len(Q)-1
                while(i<=rear):
                        print(Q[i],"<==",end=' ')
                        i+=1
                print()
```

This function will return the Front item from the Queue

This function will display Queue items

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

PYTHON CODE: QUEUE

```
Q=[]                #Queue
front=rear=None
while True:
        print("**** QUEUE DEMONSTRATION ******")
        print("1. ENQUEUE ")
        print("2. DEQUEUE")
        print("3. PEEK")
        print("4. SHOW QUEUE ")
        print("0. EXIT")
        ch = int(input("Enter your choice :"))
        if ch==1:
                val = int(input("Enter Item to Insert :"))
                Enqueue(Q,val)
```

Displaying menu to user to interact

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

PYTHON CODE: QUEUE

```
elif ch==2:
        val = Dequeue(Q)
        if val=="Underflow":
                print("Queue is Empty")
        else:
                print("\nDeleted Item was :",val)
elif ch==3:
        val = Peek(Q)
        if val=="Underflow":
                print("Queue Empty")
        else:
                print("Top Item :",val)
elif ch==4:
        Show(Q)
elif ch==0:
        print("Bye")
        break
```

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR

VARIATIONS OF QUEUE

- Circular Queue : it is implemented in circular form rather than straight line. It is used in many programming language to overcome the problems of linear queue (utilization of unused position in the beginning).
- Dequeue (Doubly-ended queue) : it is the form of queue in which elements can be added or removed from any end but not in the middle. It is of 2 type: (1) Input restricted (2) Output restricted
  - Input restricted means insertion only at one end but deletion from both end
  - Output restricted means deletion from one end and insertion from both end

VINOD KUMAR VERMA, PGT(CS), KV OEF KANPUR &
SACHIN BHARDWAJ, PGT(CS), KV NO.1 TEZPUR