

1 Queue Data Structure

2 Classification of Data Structures

3 QUEUE

4 QUEUE

5 A Print Queue in Windows

Study Point

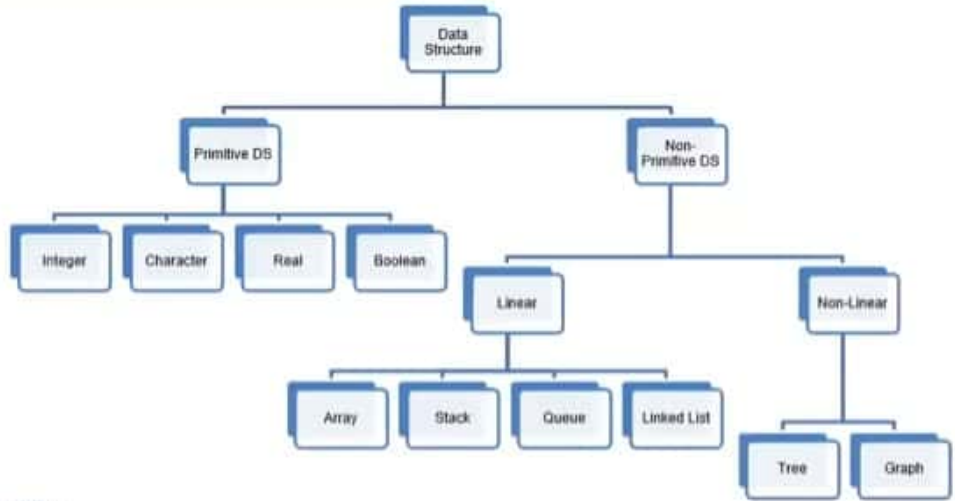
# Queue Data Structure

Presented By:

Sh. Muhammad Aamir  
Lecturer  
Department of Computer Science  
GC University, Faisalabad

- 1 Queue Data Structure
- 2 Classification of Data Structures
- 3 QUEUE
- 4 QUEUE
- 5 A Print Queue in Windows

# Classification of Data Structures



Presented By:  
Sheikh Muhammad Aamir

Click to add notes

1 Queue Data Structure

2 Classification of Data Structures

3 QUEUE

4 QUEUE

5 A Print Queue in Windows

# QUEUE




- A Queue is a Non – Primitive Linear Data Structure
- It is a linear list of elements whose elements are added at one end (the **Rear** or **Tail** of the queue) and removed from other end (the **Front** or **Head** of the queue)
- Queues have two ends:
  - Elements are added at one end called Rear or Tail.
  - Elements are removed from the other end called Front or Head.
- The element added first is also removed first i.e., **A Queue is a First In First Out (FIFO) data structure**

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

- 1 Queue Data Structure
- 2 Classification of Data Structures
- 3 QUEUE
  - It is a linear structure in which elements are added at one end and removed at the other end.
  - It is a linear structure in which elements are added at one end and removed at the other end.
  - It is a linear structure in which elements are added at one end and removed at the other end.
- 4 QUEUE
- 5 A Print Queue in Windows

# QUEUE



Rear of Queue

Front of Queue

next...

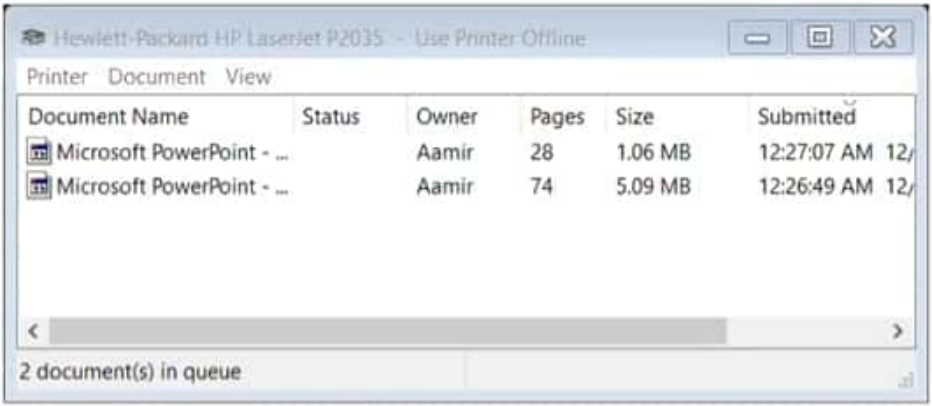

PAY HERE

Presented By:  
Sheikh Muhammad Aamir



- 3 queue
- 4 QUEUE
- 5 A Print Queue in Windows
- 6 Uses of Queues in Computing
- 7 Operations on Queue

# A Print Queue in Windows



Document Name	Status	Owner	Pages	Size	Submitted
Microsoft PowerPoint - ...		Aamir	28	1.06 MB	12:27:07 AM 12/
Microsoft PowerPoint - ...		Aamir	74	5.09 MB	12:26:49 AM 12/

2 document(s) in queue

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

3

4

5

6

7

QUEUE

QUEUE


A Print Queue in Windows

Uses of Queues in Computing

Operations on Queue

6

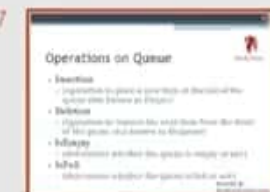
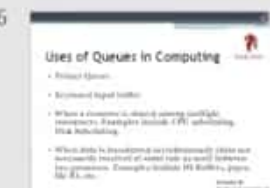
# Uses of Queues in Computing

 Study Point


- Printer Queue
- Keyboard input buffer
- When a resource is shared among multiple consumers. Examples include CPU scheduling, Disk Scheduling.
- When data is transferred asynchronously (data not necessarily received at same rate as sent) between two processes. Examples include IO Buffers, pipes, file IO, etc.

Presented By:  
Sheikh Muhammad Aamir

File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF



# Operations on Queue

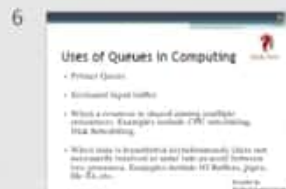
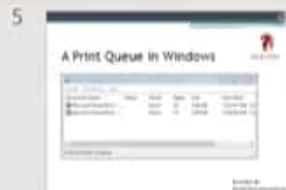
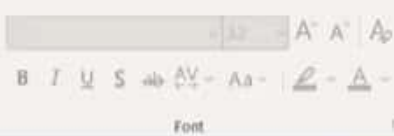
 Study Point

- **Insertion**
  - (operation to place a new item at the tail of the queue also known as Enqueue)
- **Deletion**
  - (operation to remove the next item from the front of the queue also known as Dequeue)
- **IsEmpty**
  - (determines whether the queue is empty or not)
- **IsFull**
  - (determines whether the queue is full or not)

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF



# Implementation of Queue

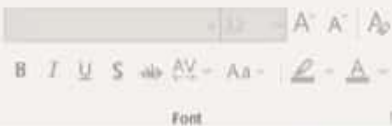
- An array-based implementation
- A linked link-based implementation
  - A linear linked list with two external references
    - A reference to the front (Front)
    - A reference to the back (Rear)

Presented By:  
Sheikh Muhammad Aamir

Click to add notes



File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF



## Operations on Queue

- Insertion
- Deletion
- Empty
- Full

## Implementation of Queue

- An array based implementation
- A linked list based implementation

## Array Representation of Queue

- Queue is maintained by a linear array, two indicators **Rear** and **Front**, that contain the location of rear element and front element of the Queue; and a variable N which represent the maximum size of array Queue.
- The **Rear = Front = 0 or NULL** will indicate that the queue is empty.

## Queue in Array



## Queue with Array



# Array Representation of Queue Study Point


- Queue is maintained by a linear array, two indicators **Rear** and **Front**, that contain the location of rear element and front element of the Queue; and a variable N which represent the maximum size of array Queue.
- The **Rear = Front = 0 or NULL** will indicate that the queue is empty.

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

- 7 Operations on Queue
- 8 Implementation of Queue
- 9 Array Representation of Queue
- 10 Queue in Array
- 11 Queue with Array

10



Study Point

# Queue in Array

Queue with size  $N = 7$

**REAR=FRONT = 0** (indicates the Queue is empty)

1	2	3	4	5	6	7

$R = 0$   
 $F = 0$

I

Presented By:  
Sheikh Muhammad Aamir

- 9 Array Representation of Queue
- 10 Queue in Array
- 11 Queue with Array
- 12 Queue with Array
- 13 Queue with Array

# Queue with Array

Study Point

Queue with size  $N = 7$

INSERT(25)

1	2	3	4	5	6	7
25						

↑↑  
F R


$R = 1$   
 $F = 1$

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

- 9 Array Representation of Queue
- 10 Queue in Array
- 11 Queue with Array
- 12 Queue with Array
- 13 Queue with Array

12



Study Point

# Queue with Array

Queue with size  $N = 7$

INSERT(50)

1	2	3	4	5	6	7
25	50					

$R = 2$

$F = 1$

$F$   $R$

Presented By:  
Sheikh Muhammad Aamir

Click to add notes



- 11 Queue with Array
- 12 Queue with Array
- 13 Queue with Array
- 14 Queue with Array
- 15 Queue with Array

13

# Queue with Array

*Study Point*

Queue with size  $N = 7$

INSERT(70)

1	2	3	4	5	6	7
25	50	70				


$R = 3$   
 $F = 1$

$\uparrow$   $\uparrow$   
 $F$   $R$

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

17



# Queue with Array

Queue with size  $N = 7$

INSERT(20)

1	2	3	4	5	6	7
25	50	70	60	35	40	20

$R = 7$   
 $F = 1$

$F$   $R$


Here Queue is full when  $F = 1$  and  $R = 7$

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

- 15 Queue with Array
- 16 Queue with Array
- 17 Queue with Array
- 18 Queue with Array
- 19 Queue with Array

18



# Queue with Array

Queue with size  $N = 7$

**DELETE()**

1	2	3	4	5	6	7
25	50	70	60	35	40	20


$R = 7$   
 $F = 2$

$F$   $R$

Presented By:  
Sheikh Muhammad Aamir

- 17 Queue with Array
- 18 Queue with Array
- 19 Queue with Array
- 20 Queue with Array
- 21 Queue with Array

20



# Queue with Array

Queue with size  $N = 7$

**DELETE()**

1	2	3	4	5	6	7
25	50	70	60	35	40	20

$R = 7$   
 $F = 4$


$F$   $R$

Presented By:  
Sheikh Muhammad Aamir



- 21 Queue with Array
- 22 Queue with Array
- 23 Queue with Array
- 24 Queue with Array
- 25 INSERT Procedure

23



# Queue with Array

Queue with size  $N = 7$

**DELETE()**

1	2	3	4	5	6	7
25	50	70	60	35	40	20

$R = 7$   
 $F = 7$

$F$   $R$

When R and F are on same index then there is only one element in Queue

Presented By:  
Sheikh Muhammad Aamir

AutoSave Off

DSA Queue Concepts · Saved to this PC

Search

aamir

Share

Comments

FileHomeInsertDesignTransitionsAnimationsSlide ShowReviewViewHelpFoxit Reader PDF

Paste

New Slide

Reuse Slides

Layout

Reset

Section

B

I

U

S

Font Color

Background Color

Font Size

Font Style

Text Direction

Align Text

Convert to SmartArt

Shapes

Arrange

Quick Styles

Shape Fill

Shape Outline

Shape Effects

Find

Replace

Select

Dictate

Design Ideas

21

Queue with Array

22

Queue with Array

23

Queue with Array

24

Queue with Array

25

INSERT Procedure

24

# Queue with Array



Queue with size  $N = 7$

DELETE()

1	2	3	4	5	6	7
25	50	70	60	35	40	20

$R = 1$   
 $F = 1$

↑

↑

F

R

Here Queue is Empty and  $F = 0$  and  $R = 0$

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

Slide 24 of 42

Notes

81%

AutoSave Off DSA Queue Concepts Search aamir

File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF

Clipboard Paste New Slide Reuse Slides Layout Reset Section Slides

Font Paragraph Drawing Editing Voice Designer

23 Queue with Array

24 Queue with Array

25 INSERT Procedure

26 DELETE Procedure

27 Queue with Array

25

# INSERT Procedure

*Study Point*

**INSERT (QUEUE, R, F, X, N)**

1. If  $F=1$  and  $R = N$  Then [Is queue already filled ?]  
Write : Queue is Full
- Else if  $R = 0$  Then  
Set  $R = F = 1$
- Else:  
Set  $R := R + 1$  [Increasing Rear by 1]  
Set  $QUEUE[R] := X$  [Insert X item at Rear side]  
[End of If structure]
2. Return

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

Slide 25 of 42

AutoSave Off DSA Queue Concepts Search aamir

File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF

Clipboard Paste New Slide Reuse Slides Layout Reset Section Slides

Font Paragraph Drawing Editing Voice Designer

23 Queue with Array

24 Queue with Array

25 INSERT Procedure

26 DELETE Procedure

27 Queue with Array

# DELETE Procedure

**DELETE (QUEUE, R, F, N)**

1. **If  $F = 0$  Then** [Is Queue already empty ?]  
    Write: Queue Empty  
    Return

**Else:**

    i. **Set  $X := \text{QUEUE}[F]$**  [Assigns Front element to X]  
    ii. **If  $F = R$  Then:**  
        Set  $F = R = 0$   
    Else:  
        Set  $F := F + 1$  [F increases by 1]  
    iii. **Return X**  
    [**end of if structure**]

**Study Point**

Presented By:  
Sheikh Muhammad Aamir


Click to add notes

Slide 26 of 42



- 25 INSERT Procedure
- 26 DELETE Procedure
- 27 Queue with Array
- 28 Circular Queue
- 29 Circular Queue

27



# Queue with Array

Queue with size  $N = 7$

1	2	3	4	5	6	7
25	50	70	60	35	40	20

F

R

Can we add a new element in this Queue or not?

Presented By:  
Sheikh Muhammad Aamir

- 25 INSERT Procedure
- 26 DELETE Procedure
- 27 Queue with Array
- 28 Circular Queue
- 29 Circular Queue

28

# Circular Queue

Study Point

Queue with size  $N = 8$

NULL or -1

F R

Presented By:  
Sheikh Muhammad Aamir

- 27 Queue with Array
- 28 Circular Queue
- 29 Circular Queue
- 30 Circular Queue
- 31 Circular Queue

29

# Circular Queue

Queue with size  $N = 8$

$R = 0$   
 $F = 0$

Study Point

Presented By:  
Sheikh Muhammad Aamir

- 27 Queue with Array
- 28 Circular Queue
- 29 Circular Queue
- 30 Circular Queue
- 31 Circular Queue

30

Study Point

# Circular Queue

Queue with size  $N = 8$

$R = 1$   
 $F = 0$

Presented By:  
Sheikh Muhammad Aamir



29

Circular Queue

30

Circular Queue

31

Circular Queue

32

Circular Queue

33

Circular Queue

31

# Circular Queue


Queue with size  $N = 8$

$R = 2$   
 $F = 0$

Presented By:  
Sheikh Muhammad Aamir

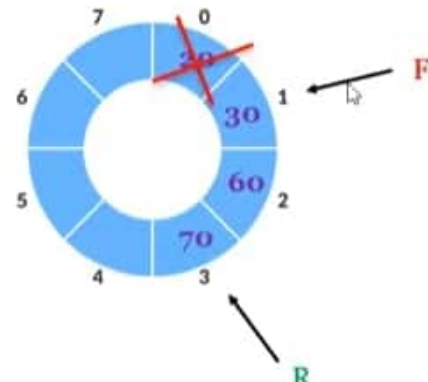
- 31 Circular Queue
- 32 Circular Queue
- 33 Circular Queue
- 34 Circular Queue
- 35 Circular Queue

33



# Circular Queue

Queue with size  $N = 8$



$R = 3$

$F = 1$

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

31 Circular Queue

32 Circular Queue

33 Circular Queue

34 Circular Queue

35 Circular Queue

34

# Circular Queue

Queue with size  $N = 8$

$R = 3$   
 $F = 2$

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

- 33 Circular Queue
- 34 Circular Queue
- 35 Circular Queue
- 36 Circular Queue
- 37 Circular Queue - Different Cases

35

# Circular Queue

Queue with size  $N = 8$

$R = 3$   
 $F = 3$

Presented By:  
Sheikh Muhammad Aamir

AutoSave Off

DSA Queue Concepts

Search

aamir

Share

Comments

FileHomeInsertDesignTransitionsAnimationsSlide ShowReviewViewHelpFoxit Reader PDF

Paste

New Slide

Reuse Slides

Layout

Reset

Section

Clipboard

Slides

Font

Paragraph

Text Direction

Align Text

Convert to SmartArt

Drawing

Shape Fill

Shape Outline

Shape Effects

Find

Replace

Select

Editing

Dictate

Design Ideas

Voice

Designer

33

Circular Queue

34

Circular Queue

35

Circular Queue

36

Circular Queue

37

Circular Queue - Different Cases

36

Circular Queue

Study Point

Queue with size  $N = 8$

7

0

1

2

3

4

5

6

$R = -1$

$F = -1$

7

0

1

2

3

4

5

6

Presented By:

Sheikh Muhammad Aamir

Click to add notes

Slide 36 of 42

Notes

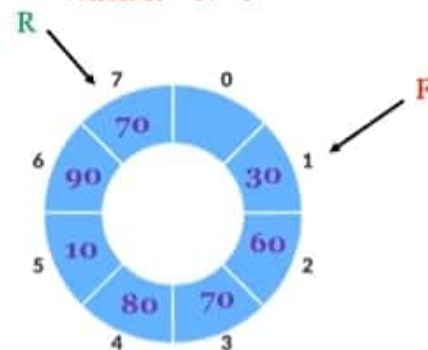
81%



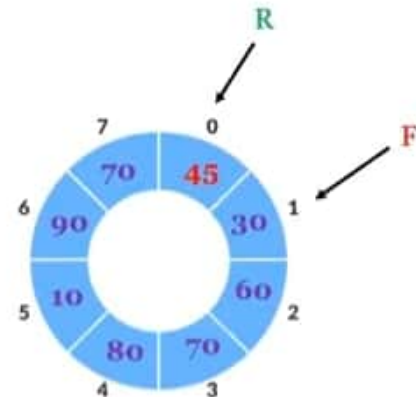
- 35 Circular Queue
- 36 Circular Queue
- 37 Circular Queue - Different Cases
- 38 Circular Queue - Different Cases
- 39 Circular Queue - Different Cases

# Circular Queue - Different Cases Study Point

Queue with size  $N = 8$   
When  $R = N - 1$



$R = 7$   
 $F = 1$



$R = 0 \ (7+1=8 \% 8 = 0)$   
 $F = 1$

Here Queue is also full when  $F = R + 1$

Presented By: Sheikh Muhammad Aamir

AutoSave Off

DSA Queue Concepts

Search

aamir

FileHomeInsertDesignTransitionsAnimationsSlide ShowReviewViewHelpFoxit Reader PDF

Paste

New Slide

Reuse Slides

Layout

Reset

Section

Font

Paragraph

Drawing

Editing

Voice

Designer

Share

Comments

35

Circular Queue

36

Circular Queue

37

Circular Queue - Different Cases

38

Circular Queue - Different Cases

39

Circular Queue - Different Cases

38

# Circular Queue - Different Cases

Study Point

Queue with size N = 8

R = 0

F = 2

Index	0	1	2	3	4	5	6	7
Value	70	45	44	60	70	80	10	90

R = 1

F = 2

Index	0	1	2	3	4	5	6	7
Value	70	45	44	60	70	80	10	90

Here Queue is also full when  $F = R + 1$

Presented By: Sheikh Muhammad Aamir

Slide 38 of 42

Notes

81%

AutoSave Off

DSA Queue Concepts

Search

aamir

Share

Comments

FileHomeInsertDesignTransitionsAnimationsSlide ShowReviewViewHelpFoxit Reader PDF

Paste

New Slide

Reuse Slides

Layout

Reset

Section

Font

Paragraph

Drawing

Editing

Voice

Designer

37

Circular Queue - Different Cases

38

Circular Queue - Different Cases

39

Circular Queue - Different Cases

40

Circular INSERT Procedure

41

Circular DELETE Procedure

39

# Circular Queue - Different Cases

Study Point

Queue with size  $N = 8$   
When  $F = N - 1$

F → 7

R → 0

$R = 0$   
 $F = 7$

R

F

$R = 0$   
 $F = 0 \ (7+1=8 \% 8 = 0)$

Presented By:  
Sheikh Muhammad Aamir

Slide 39 of 42

Notes

81%

AutoSave Off DSA Queue Concepts Search aamir

File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF

Paste New Slide Reuse Slides Layout Reset Section Slides

Font Paragraph Drawing Editing Voice Designer

37 Circular Queue - Different Cases

38 Circular Queue - Different Cases

39 Circular Queue - Different Cases

40 Circular INSERT Procedure


41 Circular DELETE Procedure

Click to add notes

Slide 40 of 42

40

# Circular INSERT Procedure

 Study Point

**INSERT (QUEUE, R, F, X, N)**

1. If  $(F=0 \text{ and } R = N - 1) \text{ or } F = R+1$  Then [Is queue already filled ?]  
Write : Queue is Full  
Else if  $R = -1$  Then  
Set  $R = F = 0$   
Else if  $R = N - 1$ :  
Set  $R := 0$   
Else:  
Set  $R := R + 1$  (or Set  $R := (R+1) \% N$ )  
[Increasing Rear by 1]  
Set  $QUEUE[R] := X$  [Insert X item at Rear side]  
[End of If structure]
2. Return

Presented By:  
Sheikh Muhammad Aamir



AutoSave Off DSA Queue Concepts Search aamir

File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF

Clipboard Paste New Slide Reuse Slides Layout Reset Section Slides

Font Paragraph Drawing Editing Voice Designer

39 Circular Queue - Different Cases

40 Circular INSERT Procedure

41 Circular DELETE Procedure

42 Implementation Assignment

# Circular DELETE Procedure

**DELETE (QUEUE, R, F, N)**

1. **If  $F = -1$  Then** [Is Queue already empty ?]  
Write: Queue Empty  
Return

**Else:**

i. **Set  $X := \text{QUEUE}[F]$**  [Assigns Front element to X]

ii. **If  $F = R$  Then:**  
Set  $F = R = 0$   
**Else if  $F = N - 1$  Then:**  
Set  $F = 0$   
**Else:**  
Set  $F := F + 1$  (or Set  $F := (F+1) \% N$ ) [F increases by 1]

iii. **Return X** [end of if structure]

Study Point

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

Slide 41 of 42



AutoSave Off DSA Queue Concepts Search aamir

File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF

Clipboard Paste New Slide Reuse Slides Layout Reset Section Slides

Font Paragraph Drawing Editing Voice Designer

39 Circular Queue - Different Cases

40 Circular INSERT Procedure

41 Circular DELETE Procedure

42 Implementation Assignment

42

# Implementation Assignment

Study Point


Write a program and check its performance in C++ or any language to implement the Queue with the help of **INSERTION** AND **DELETION** operations.

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

Slide 42 of 42

- 1 Double Ended Queue - Deque
- 2 Double-ended Queue (Deque)
- 3 Double-ended Queue (Deque)
- 4 Double-ended Queue (Deque)
- 5 Double-ended Queue (Deque)



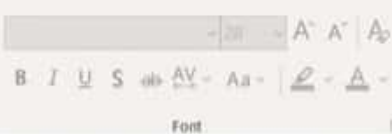
Study Point

# Double Ended Queue - Deque

Presented By:

Sh. Muhammad Aamir  
 Lecturer  
 Department of Computer Science  
 GC University, Faisalabad

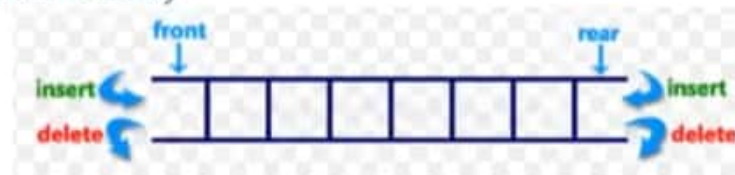
Click to add notes



# Double-ended Queue (Deque)

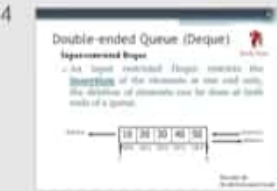
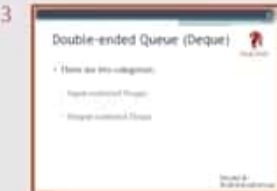
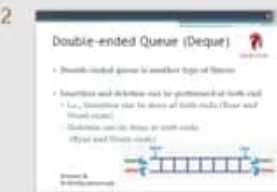
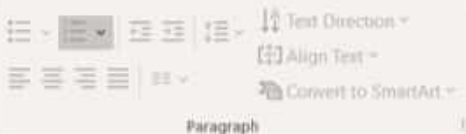


- Double ended queue is another type of Queue
- Insertion and deletion can be performed at both end
  - i.e., Insertion can be done at both ends (Rear and Front ends)
  - Deletion can be done at both ends (Rear and Front ends)



Presented By:  
Sheikh Muhammad Aamir

Click to add notes



# Double-ended Queue (Deque)

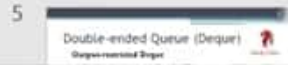
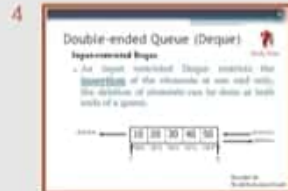
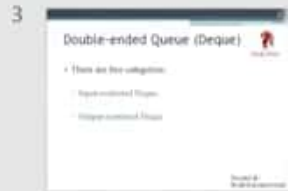
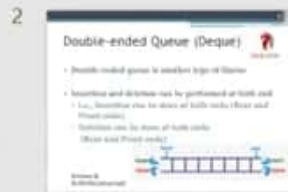


- There are two categories:
  - Input-restricted Deque.
  - Output-restricted Deque.

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF

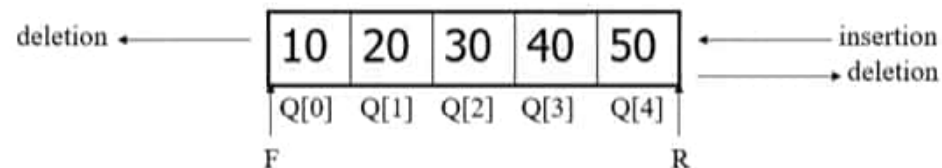


# Double-ended Queue (Deque)

*Study Point*

## Input-restricted Deque

- An input restricted Deque restricts the insertion of the elements at one end only, the deletion of elements can be done at both ends of a queue.



Presented By:  
Sheikh Muhammad Aamir

Click to add notes



File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF

Clipboard

Paste  
New Slide  
Reuse Slides  
Layout  
Reset  
Section

Slides

Font  
B I U S A V A A -  
Paragraph  
Text Direction  
Align Text  
Convert to SmartArt

Paragraph

Drawing  
Arrange  
Quick Styles  
Shape Fill  
Shape Outline  
Shape Effects

Drawing

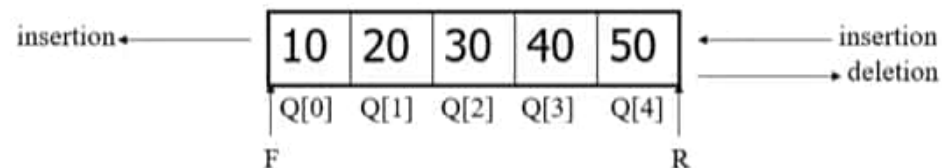
Editing  
Find  
Replace  
SelectVoice  
Dictate  
Design Ideas  
Designer

# Double-ended Queue (Deque)

*Study Point*

## Output-restricted Deque

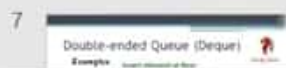
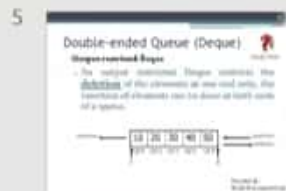
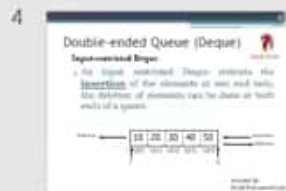
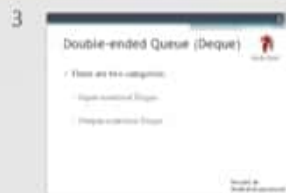
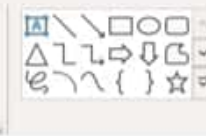
- An output restricted Deque restricts the **deletion** of the elements at one end only, the insertion of elements can be done at both ends of a queue.



Presented By:  
Sheikh Muhammad Aamir

Click to add notes

File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF



# Double-ended Queue (Deque)

**Input-restricted Deque Functions**

1. Insertion from Rear side
2. **Deletion from Rear side**
3. Deletion from Front side

**Output-restricted Deque Functions**

1. Insertion from Rear side
2. **Insertion from Front side**
3. Deletion from Front side

Presented By:  
Sheikh Muhammad Aamir



Click to add notes

Notes



5 Double-ended Queue (Deque) Important Deque Functions

6 Double-ended Queue (Deque) Important Deque Functions

7 Double-ended Queue (Deque) Example

8 Double-ended Queue (Deque) Example

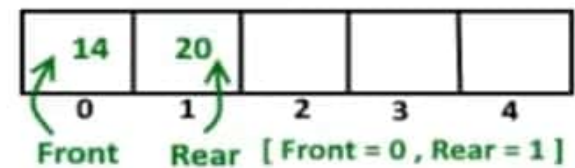
9 Insertion from Rear side

# Double-ended Queue (Deque)

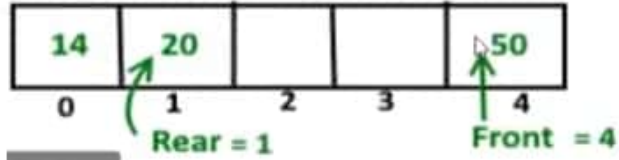


Examples:

Insert element at Rear



Insert element at Front end  
Now Front points last index



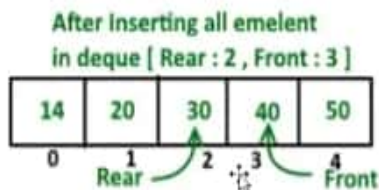
Presented By:  
Sheikh Muhammad Aamir

Click to add notes

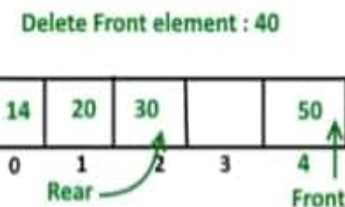
- 5 Double-ended Queue (Deque) Supercharged Slides
- 6 Double-ended Queue (Deque) Supercharged Slides
- 7 Double-ended Queue (Deque) Example
- 8 Double-ended Queue (Deque) Example
- 9 Insertion from Rear side

# Double-ended Queue (Deque)

## Examples:



Delete Element from Front end , New front

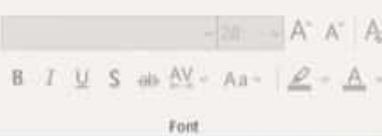


Presented By:  
Sheikh Muhammad Aamir

Click to add notes



File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF



## Insertion from Rear side



### INSERT\_REAR (QUEUE, R, F, X, N)

1. If  $(F=0 \text{ and } R = N - 1) \text{ or } F = R+1$  Then [Is queue already filled?]

Write : Queue is Full

Else if  $R = -1$  Then [first time insertion]

Set  $R = F = 0$

Else if  $R = N - 1$ :

Set  $R := 0$

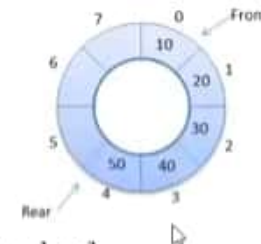
Else:

Set  $R := R + 1$

Set  $QUEUE[R] := X$

[End of If structure]

2. Return



[Increasing Rear by 1]

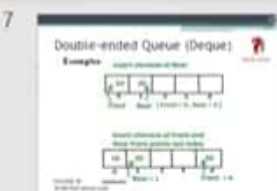
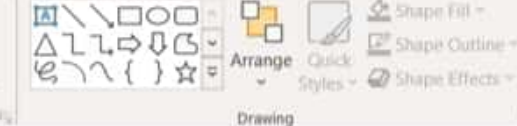
[Insert X item at Rear side]

Presented By:  
Sheikh Muhammad Aamir

Click to add notes



File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF



# Insertion from Front side



## INSERT\_FRONT (QUEUE, R, F, X, N)

1. If  $(R=0 \text{ and } F = N - 1) \text{ or } R = F + 1$  Then [Is queue already filled?]

Write : Queue is Full

Else if  $F = -1$  Then [first time insertion]

Set  $F=R=0$

Else if  $F = 0$ :

Set  $F := N - 1$

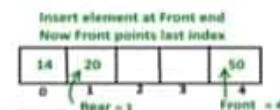
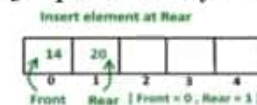
Else:

Set  $F := F - 1$

Set  $QUEUE[F] := X$

[End of If structure]

2. Return

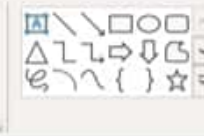
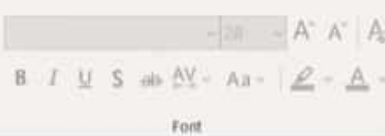


[Decreases Front by 1]  
[Insert X item at Front side]

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF



# Deletion from Front side



## DELETE\_FRONT (QUEUE, R, F, N)

1. If  $F = -1$  Then

[Is Queue already empty ?]

Write: Queue Empty  
Return

Else:

i. Set  $X := \text{QUEUE}[F]$

[Assigns Front element to X]

ii. If  $F = R$  Then:

[only one element in queue]

Set  $F = R = -1$

Else if  $F = N - 1$  Then:

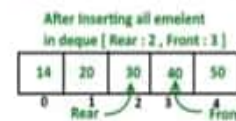
Set  $F = 0$

Else:

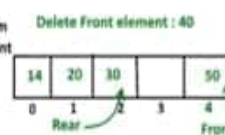
Set  $F := F + 1$

[F increases by 1]

iii. Return X  
[end of if structure]



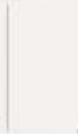
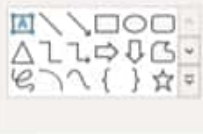
Delete Element from Front end, New front



Click to add notes

Notes



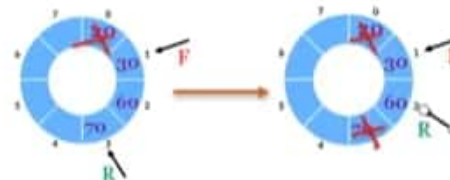


# Deletion from Rear side



## DELETE\_REAR (QUEUE, R, F, N)

1. If  $R = -1$  Then [Is Queue already empty ?]  
 Write: Queue Empty, so cannot delete from Rear end  
 Return  
 Else:
    - i. Set  $X := \text{QUEUE}[R]$  [Assigns Rear element to X]
    - ii. If  $F = R$  Then:  
 Set  $F = R = -1$   
 Else if  $R = 0$  Then:  
 Set  $R = N - 1$   
 Else:  
 Set  $R := R - 1$
    - iii. Return X
- [end of if structure]



Presented By:  
Sheikh Muhammad Aamir

Click to add notes

AutoSave Off DSA Double Ended Queue by Aamir Sheikh - Saved to this PC Search aamir

File Home Insert Design Transitions Animations Slide Show Review View Help Foxit Reader PDF

Clipboard Paste New Slide Reuse Slides Layout Reset Section Slides

Font Paragraph Drawing Editing Voice Designer

10 Insertion from Front side

11 Deletion from Front side

12 Deletion from Rear side

13 Implementation Assignment

# Implementation Assignment

Study Point

Write a program and check its performance in C++ or any language to implement the Deque with the help of **INSERTION** AND **DELETION** operations.

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

Slide 12 of 13