

AutoSave Off DSA - Linked List by Aamir... 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

1 Linked List

2 Classification of Data Structures

3 Linked List

4 Linked List

5 Linked List

Click to add notes

Study Point

# Linked List

Presented By:

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

Slide 1 of 27 Recovered

- 1
- 2
- 3
- 4
- 5

2

# Classification of Data Structures

```

graph TD
    DS[Data Structure] --> PDS[Primitive DS]
    DS --> NPS[Non-Primitive DS]
    PDS --> Integer
    PDS --> Character
    PDS --> Real
    PDS --> Boolean
    NPS --> Linear
    NPS --> Non-Linear
    Linear --> Array
    Linear --> Stack
    Linear --> Queue
    Linear --> Linked List
    Non-Linear --> Tree
    Non-Linear --> Graph
    
```

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

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



# Linked List



- Non-Primitive Linear Data Structure
- List of elements that are linked through some addresses
- Collection of nodes
- Each node contains two parts i.e.,
  - Data/Information part
  - Address/Link part



node

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

AutoSave Off

DSA - Linked List by Aamir...

Search

aamir

Share

Comments

FileHomeInsertDesignTransitionsAnimationsSlide ShowReviewViewHelpFoxit Reader PDF

Paste

New Slide

Reuse Slides

Layout

Reset

Section

B I U S

Font

Text Direction

Align Text

Convert to SmartArt

Paragraph

Shape Fill

Shape Outline

Shape Effects

Arrange

Quick Styles

Drawing

Find

Replace

Select

Editing

Dictate

Design Ideas

Voice

Designer

1

2

3

4

5

LinkedList

Classification of Data Structures


LinkedList

LinkedList

LinkedList

4

LinkedList



Study Point

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

Notes

81%

- 3 Linked List
- 4 Linked List
- 5 Linked List
- 6 Why we use Linked List?
- 7 Linked List : Insertion and Deletion

5

# Linked List

*Study Point*

LinkedList

head

node

Data

Next

node

Data

Next

node

Data

Next

Null

Head node contains the address of first node

If head node is NULL then there is no linked list

Other nodes contain the address of next node

Last node in the linked list contains NULL in next part

Presented By:  
Sheikh Muhammad Aamir

Click to add notes





# Why we use Linked List?



- Array has a fixed size (a limited number of elements)
  - Data must be shifted during insertions and deletions
- Can partially solve this problem by reallocating the array as needed (how much memory to add?)
- Linked list is able to grow in size as needed (dynamically memory allocation)
  - Does not require the shifting of items during insertions and deletions, so insertion and deletion are efficient

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

AutoSave Off

DSA - Linked List by Aamir...

Search

aamir

Share

Comments

FileHomeInsertDesignTransitionsAnimationsSlide ShowReviewViewHelpFoxit Reader PDF

Paste

New Slide

Reuse Slides

Layout

Reset

Section

B I U S

Font

Text Direction

Align Text

Convert to SmartArt

Shape Fill

Shape Outline

Shape Effects

Find

Replace

Select

Dictate

Design Ideas

Clipboard

Slides

Font

Paragraph

Drawing

Editing

Voice

Designer

5

6

7

8

9

Linked List

Why we use Linked List?

Linked List : Insertion and Deletion

Pointers

Pointers

7

Linked List : Insertion and Deletion

(a) 20 → 45 → 51 → 76 → 84

(b) 20 → 45 → 51 → 60 → 76 → 84

(c) 20 → 45 → 51 → 60 → 76 → 84

Figure a) A linked list of integers; b) insertion; c) deletion

Presented By: Sheikh Muhammad Aamir

Click to add notes

Slide 7 of 27

Recovered

81%

# Pointers



- A pointer is a variable that contains the address or memory location of another variable
  - A statically allocated pointer declaration  
`int *p;`
  - A dynamically allocated pointer variable  
`p = new int;`
- By using **new** operator, a space (as per type mentioned) is created in memory and **address** of that memory is passed to a pointer

Presented By:  
Sheikh Muhammad Aamir

Click to add notes



7 Linked List: Insertion and Deletion

8 Pointers

- A pointer is a variable that contains the address of another variable.
- A variable declared as a pointer variable, and its value is the address of another variable.
- By using a pointer variable, we can access the memory of another variable.

9 Pointers

- The & operator places the address of a variable into a pointer variable.
- The \* operator is also known as reference operator.
- The expression \*p, denotes the memory cell to which p points.
- \* is known as de-reference operator.

10 Pointers

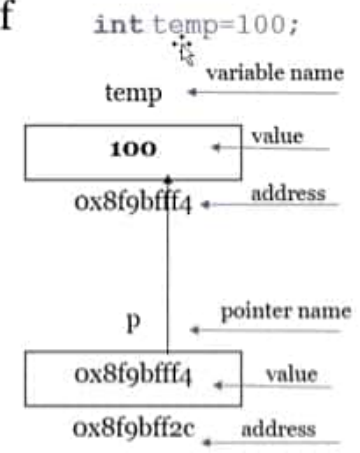
- The & operator places the address of a variable into a pointer variable.
- The \* operator is also known as reference operator.
- The expression \*p, denotes the memory cell to which p points.
- \* is known as de-reference operator.

11 Pointers

# Pointers



- The **&** operator places the address of a variable into a pointer variable
  - `p = &temp;`
- &** operator is also known as reference operator
- The expression **\*p**, denotes the memory cell to which **p** points
- \*** is known as de-reference operator



Presented By:  
Sheikh Muhammad Aamir

Click to add notes

AutoSave Off DSA - Linked List by Aamir... 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

7 8 9 10 11

Linked List: Insertion and Deletion

Pointers

- A pointer is a variable that contains the address of another location of memory variable.
- A dynamically allocated pointer declaration, e.g., `int *p;`
- A dynamically allocated pointer variable, e.g., `int *p; new int;`
- By using `&` operator, a pointer (e.g., `int *p;`) is created in memory and address of that memory is passed to a pointer.

Pointers

- This is operator gives the address of a variable into a pointer variable. `p = &100;`
- `&` operator is also known as reference operator.
- The expression `*p` denotes the memory cell to which `p` points.
- `*p` is known as the reference expression.

Pointers

- The `delete` operator returns dynamically allocated memory to the system for reuse, and leaves the variable undefined.
- `delete p;`
- A pointer to a deallocated memory cell is possible and dangerous.
- Assign the pointer `q` the value in `p`.  
`q = p;`


Presented By:  
Sheikh Muhammad Aamir

Click to add notes

Slide 10 of 27 Recovered

- 9 Pointers
- 10 Pointers
- 11 Pointers
- 12 Pointers
- 13 Types of Linked List

# Pointers



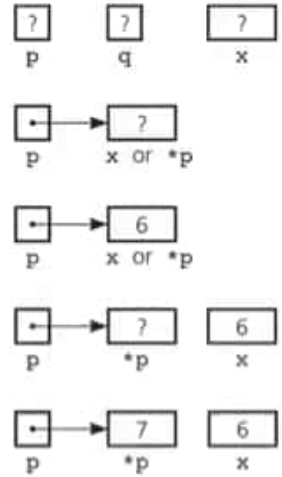
(a) `int *p, *q;`  
`int x;`

(b) `p = &x;`

(c) `*p = 6;`

(d) `p = new int;`

(e) `*p = 7;`



(a) declaring pointer variables;  
(b) pointing to statically allocating memory;  
(c) assigning a value,  
(d) allocating memory dynamically;  
(e) assigning a value

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

- 9 Pointers
- 10 Pointers
- 11 Pointers
- 12 Pointers
- 13 Types of Linked List

# Pointers

(f) `q = p;`

(g) `q = new int;`  
`*q = 8;`

(h) `p = NULL;`

(i) `delete q;`  
`q = NULL;`

12

**Study Point**

(f) copying a pointer;  
(g) allocating memory dynamically and assigning a value;  
(h) assigning NULL to a pointer variable;  
(i) deallocating memory

Presented By:  
Sheikh Muhammad Aamir

Click to add notes



12 Pointers

13 Types of Linked List

14 Singly Linked List

15 Linked List Operations

16 Linked List Operations...

# Types of Linked List

**Study Point**

- There are following types of linked list
  - Singly Linked List** - Item navigation is forward only.
  - Doubly Linked List** - Items can be navigated forward and backward.
  - Circular Linked List** - Last item contains link of the first element as next and the first element has a link to the last element as previous.

D = Data P = Previous N = Next		
<b>1) Singly Linked List</b> 	<b>2) Doubly Linked List</b> 	<b>3) Circular Linked List</b> 

Presented By:  
Sheikh Muhammad Aamir

AutoSave Off DSA - Linked List by Aamir... Search aamir

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

Paste New Slide Reuse Slides Layout Reset Section

Clipboard Slides

Font Paragraph Drawing Editing Voice Designer

12 Pointers

13 Types of Linked List

14 Singly Linked List

15 Linked List Operations

16 Linked List Operations...

Click to add notes

Slide 14 of 27 Recovered

# Singly Linked List

Study Point

- Uni-directional/one-way linked list
- A single node contains data and a pointer to the next node which helps in maintaining the structure of the list.
- A node in a single linked list is usually structured as

```
struct Node
{
    int Data;
    Node *Next;
};
```
- A node is dynamically allocated

```
Node *p;
p = new Node;
```
- Reference a node member with the -> operator  
i.e. `p->data;`  
`p-> Next`

Presented By:  
Sheikh Muhammad Aamir

Diagram illustrating a node structure: `Data` | `Next` → `Data` | `Next`

AutoSave Off DSA - Linked List by Aamir... 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

12 Pointers

13 Types of Linked List


14 Singly Linked List

15 Linked List Operations

16 Linked List Operations...

15

# Linked List Operations

 Study Point

- Traversing/Display the linked list
- Insertion of a new node
  - Insertion at front
  - Insertion at end
  - Insertion in middle
    - (After a particular value / or particular node)
- Deletion of a node
  - Deletion of first node
  - Deletion of Last node
  - Deletion from middle
    - (A particular node by value or by node number)

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

Slide 15 of 27 Recovered

AutoSave Off DSA - Linked List by Aamir... 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

14 Singly Linked List

15 Linked List Operations


16 Linked List Operations...

17 Singly Linked List Program understanding

18 Singly Linked List Program understanding

16

# Linked List Operations...



- Searching a value(s) from linked list
- Sorting the linked list
- Merging two linked lists

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

Slide 16 of 27 Recovered



AutoSave Off DSA - Linked List by Aamir... 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

14 Singly Linked List

15 Linked List Operations

16 Linked List Operations...

17 Singly Linked List Program Understanding

18 Singly Linked List Program Understanding

## Singly Linked List Program Understanding

Study Point

```
#include<iostream>
using namespace std;
struct Node
{ int data
  Node *next;
}*p, *q, *head, *temp;
void createnode(void);
void displaylist(void);
int count = 0;
int main( )
{ int choice;
  char more;
  head = NULL;
  do
  { cout<<"Press 1 to Insert a new node";
    cout<<"Press 2 to show the linked list";
    cout<<"Enter your Choice : ";
    cin>> choice;
    if (choice == 1)
      createnode( ); // so on
```

data	next
------	------

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

Slide 17 of 27 Recovered

AutoSave Off DSA - Linked List by Aamir... 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

16 Linked List Operations...

17 Singly Linked List Program Understanding

18 Singly Linked List Program Understanding

19 Singly Linked List Program Understanding

20 Singly Linked List Program Understanding

# Singly Linked List Program Understanding

```
int main( )
{ int choice;
  char more;
  head = NULL;
  do
  { cout<<"Press 1 to Insert a new node";
    cout<<"Press 2 to show the linked list";
    cout<<"Enter your Choice : ";
    cin>> choice;
    if (choice == 1)
      createnode( );
    else if(choice == 2)
      displaylist( );

    *I
    *
    *
```

Study Point

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

Slide 18 of 27 Recovered

AutoSave Off DSA - Linked List by Aamir... 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

16 Linked List Operations...

17 Singly Linked List Program Understanding

18 Singly Linked List Program Understanding

19 Singly Linked List Program Understanding

20 Singly Linked List Program Understanding

19

```
void createnode( )
{
    p = new Node;
    count++; //to count the number of nodes
    cout<<"Enter value for node : ";
    cin>>p->data;
    p->next = NULL;

    if(head == NULL)
        head = p;
    else
    {
        temp = head;
        while(temp->next != NULL)
        {
            temp = temp->next;
        }
        temp->next = p;
    }
}
```

Study Point

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

Slide 19 of 27 Recovered

- 18 Singly Linked List Program Understanding
- 19 Singly Linked List Program Understanding
- 20 Singly Linked List Program Understanding
- 21 Singly Linked List Program Understanding
- 22 Singly Linked List Program Understanding

20

**head = NULL;**  
**createnode( );**    **//first call**

p = 0x7f100  
p->data = 20  
p->next = NULL

p	
20	NULL
0x7f100	

As first time head = NULL, so we assign p to head i.e. head = p (head = 0x7f100)

head

p

20 | NULL  
0x7f100

head = 0x7f100  
p = 0x7f100

**Study Point**

Presented By:  
Sheikh Muhammad Aamir



AutoSave Off

DSA - Linked List by Aamir...

Search

aamir

Share

Comments

FileHomeInsertDesignTransitionsAnimationsSlide ShowReviewViewHelpFoxit Reader PDF

Paste

New Slide

Reuse Slides

Layout

Reset

Section

Font

Paragraph

Drawing

Editing

Voice

Designer

Find

Replace

Select

Dictate

Design Ideas

18

21

22

Slide 18: Singly Linked List Program Understanding

Slide 19: void insertAtEnd()

Slide 20: head = NULL, createNode()

Slide 21: head = NULL, createNode()

Slide 22: Diagram of linked list

head = 0x7f100;  
createnode( ); //second call

p = 0x7f200  
p->data = 30  
p->next = NULL

p

30 | NULL

0x7f200

As head != NULL, so we will adopt else part

else  
{ temp = head;  
while(temp->next != NULL)  
{ temp = temp->next;  
}  
temp->next = p;  
}

head = 0x7f100  
p = 0x7f100  
p = 0x7f200  
temp = 0x7f100

Presented By:  
Sheikh Muhammad Aamir

Click to add notes

Slide 21 of 27

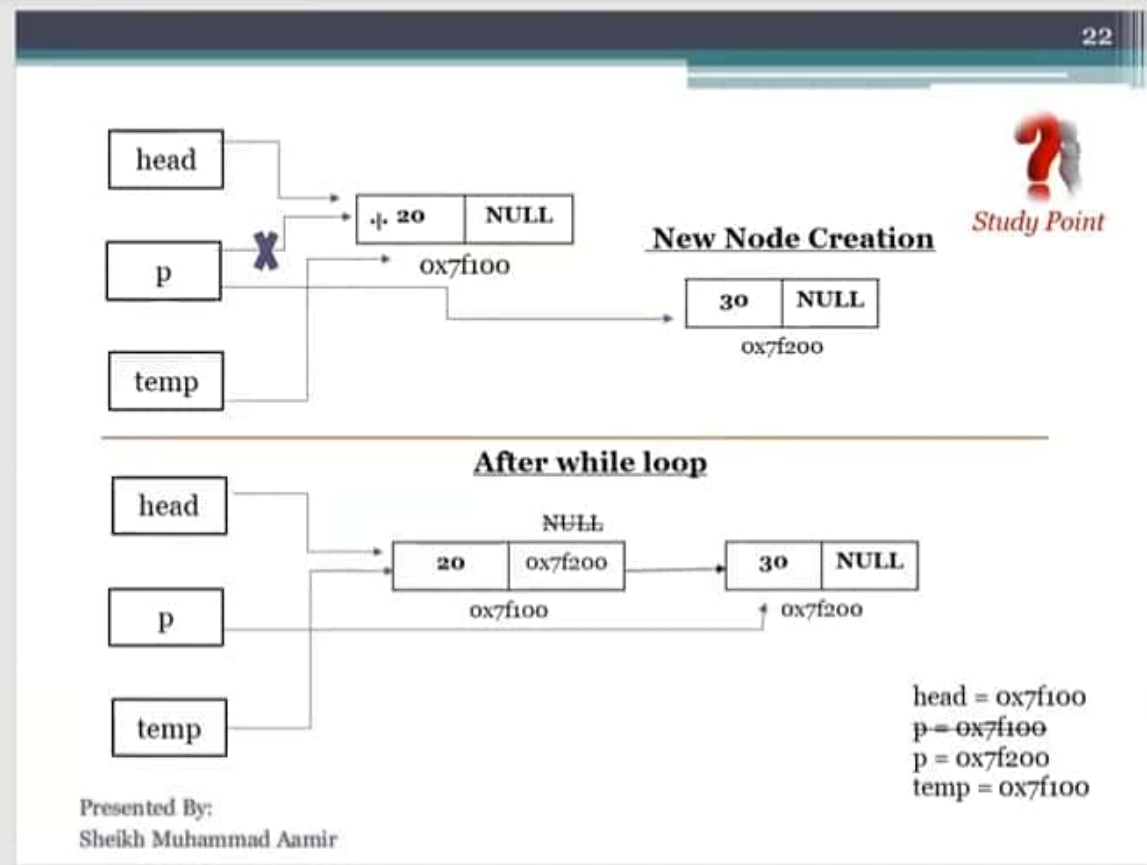
Recovered

Notes

81%

Scanned with CamScanner

- 20
- 21
- 22
- 23
- 24



Presented By:  
Sheikh Muhammad Aamir

AutoSave Off DSA - Linked List by Aamir... 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

20 21 22 23 24

Click to add notes

Slide 23 of 27 Recovered

23

**head = 0x7f100;**  
**createnode( );**      **//third call**

**p = 0x7f300**  
**p->data = 50**  
**p->next = NULL**

p	
50	NULL
0x7f300	

Study Point

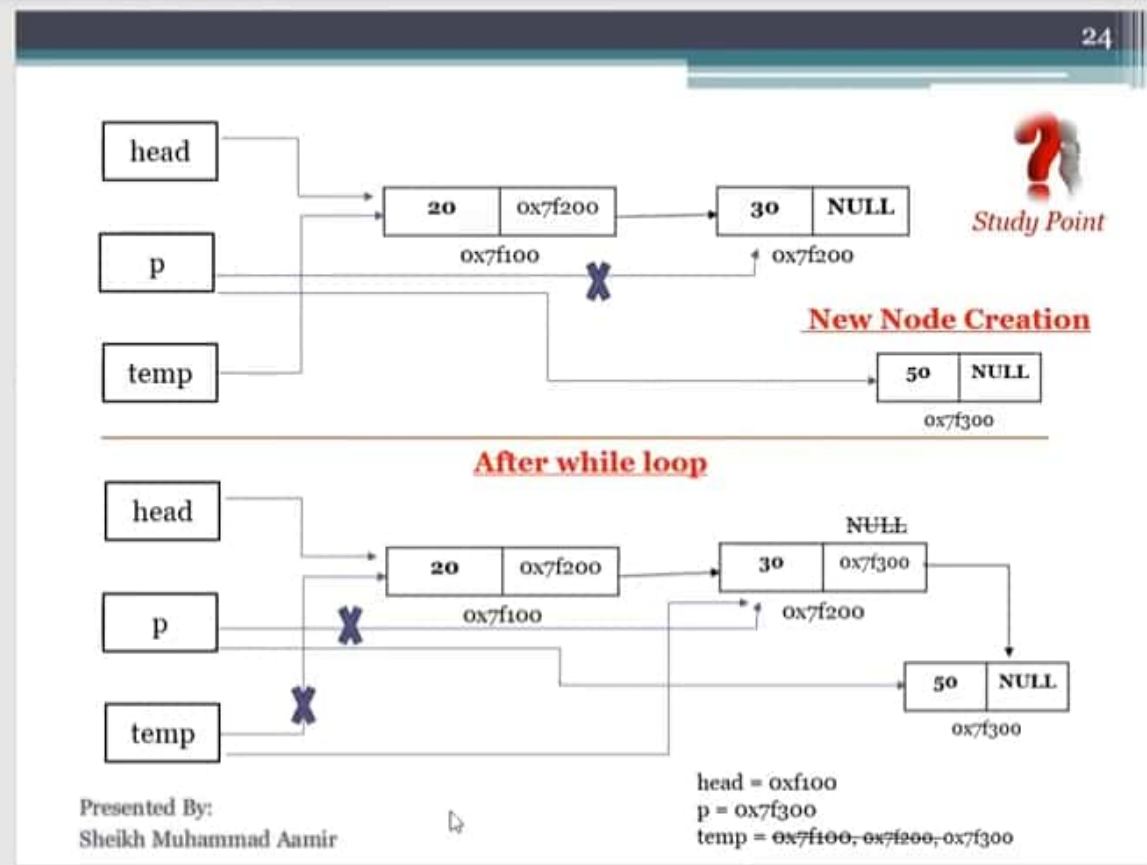
As head != NULL, so we will adopt else part

```
else
{
    temp = head;
    while(temp->next != NULL)
    {
        temp = temp->next;
    }
    temp->next = p;
}
```

head = 0x7f100  
p = 0x7f100  
p = 0x7f200  
p = 0x7f300  
temp = 0x7f100

Presented By:  
Sheikh Muhammad Aamir

- 22
- 23
- 24
- 25
- 26



Click to add notes



AutoSave Off DSA - Linked List by Aamir... 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

22 23 24 25 26

Click to add notes

Slide 25 of 27 Recovered

25

**Study Point**

```
void displaylist( )
{ if(head!=NULL || count!=0)
  {
    temp = head;
    while(temp!=NULL)
    { cout<<temp->data;
      temp=temp->next;
    }
  }
  else
  { cout<<"The linked list is not created yet";
  }
}
```

head → 20 | 0x7f200 → 30 | 0x7f300 → 50 | NULL

0x7f100 0x7f200 0x7f300

Presented By:  
Sheikh Muhammad Aamir