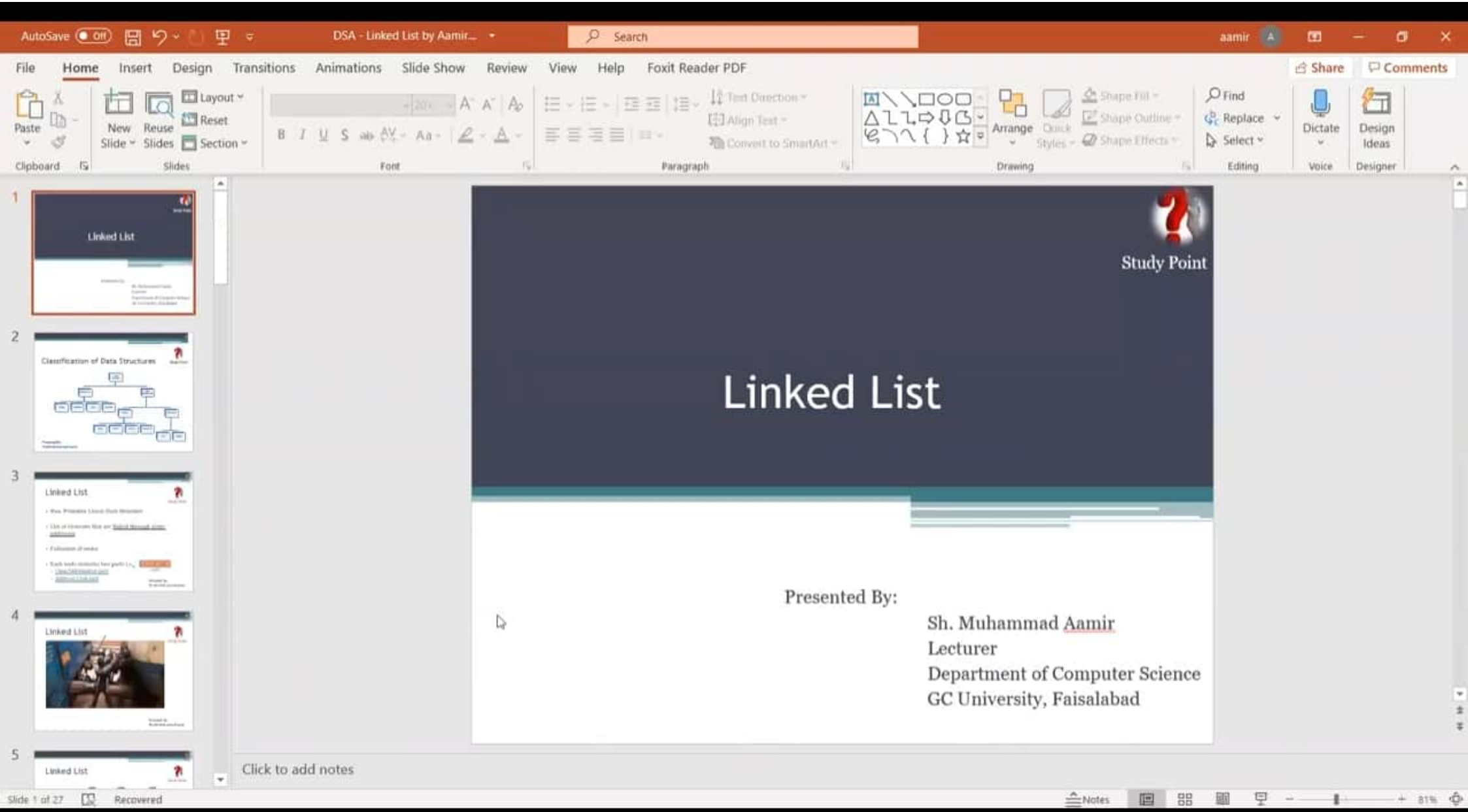


AutoSave Off | DSA - Linked List by Aamir... | Search | aamir

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

Paste | New Slide | Reuse Slides | Layout | Reset | Section | Clipboard | Slides | Font | Paragraph | Drawing | Arrange | Quick Styles | Shape Fill | Shape Outline | Shape Effects | Find | Replace | Select | Editing | Dictate | Voice | Design Ideas | Designer

1 Linked List

2 Classification of Data Structures

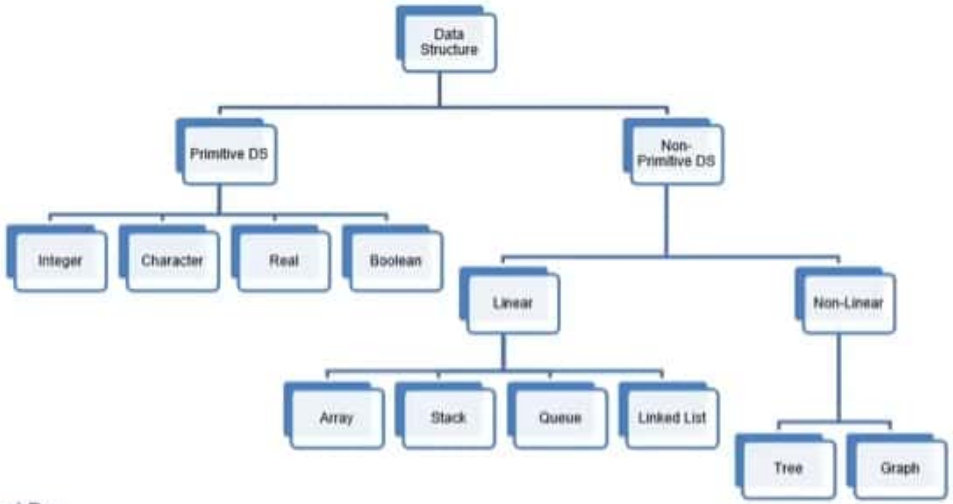
3 Linked List

4 Linked List

5 Linked List

Classification of Data Structures

Study Point



Presented By:
Sheikh Muhammad Aamir

Click to add notes

Slide 2 of 27 | Recovered | Notes | 81%

Linked List

Study Point

- 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

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

Paste | New Slide | Reuse Slides | Layout | Reset | Section | Clipboard | Slides | Font | Paragraph | Drawing | Arrange | Quick Styles | Shape Fill | Shape Outline | Shape Effects | Find | Replace | Select | Editing | Dictate | Voice | Design Ideas | Designer

1 Linked List

2 Classification of Data Structures

3 Linked List

4 Linked List

5 Linked List

Linked List

Study Point

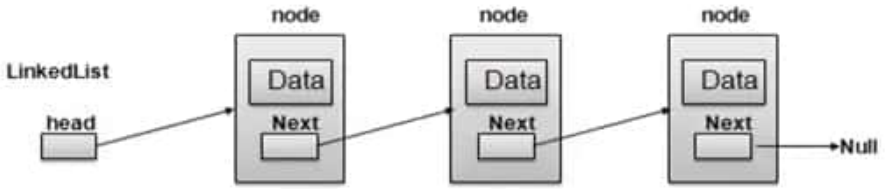
Presented By:
Sheikh Muhammad Aamir

Click to add notes

Slide 4 of 27 | Recovered | Notes | 81%

Linked List

Study Point



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

Why we use Linked List?

Study Point

- 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

Linked List : Insertion and Deletion

Study Point

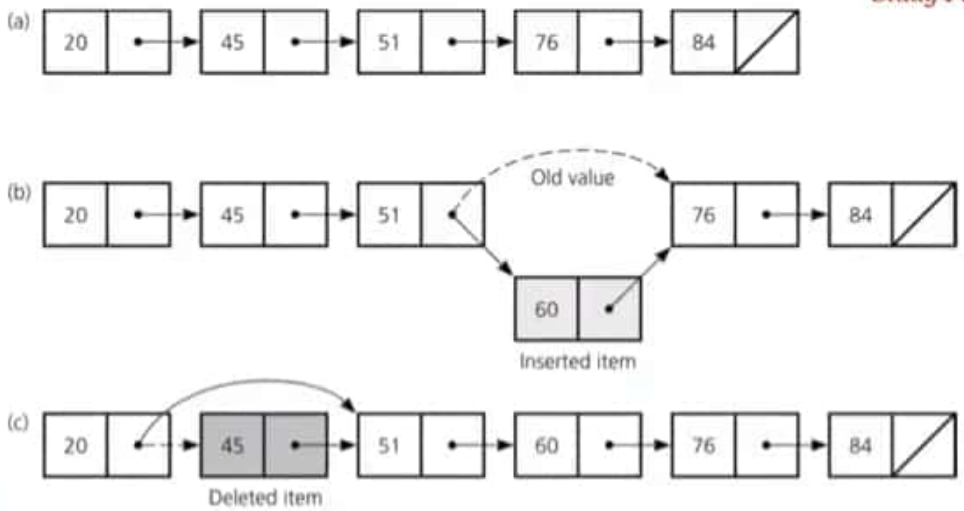


Figure a) A linked list of integers; b) insertion; c) deletion

Presented By:
Sheikh Muhammad Aamir

Pointers

Study Point

- 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

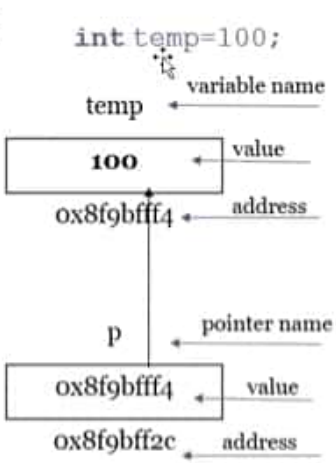
Presented By:
Sheikh Muhammad Aamir

Click to add notes

Pointers

Study Point

- 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

Pointers

Study Point

- 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;
```

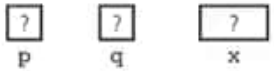
Presented By:
Sheikh Muhammad Aamir

Click to add notes

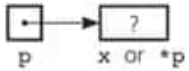
Pointers

Study Point

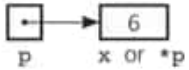
(a) `int *p, *q;`
`int x;`



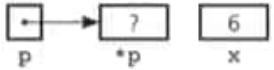
(b) `p = &x;`



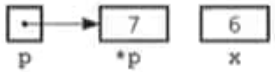
(c) `*p = 6;`



(d) `p = new int;`



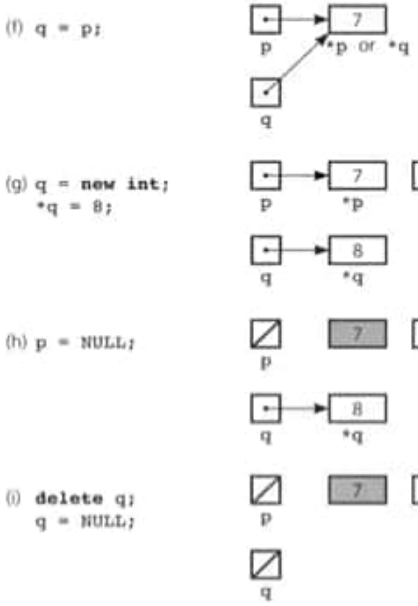
(e) `*p = 7;`



(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

Pointers



Study Point

(f) copying a pointer;

(g) allocating memory dynamically and assigning a value;

(h) assigning NULL to a pointer variable;

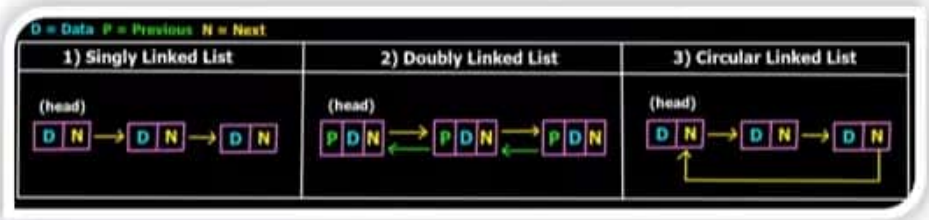
(i) deallocating memory

Presented By:
Sheikh Muhammad Aamir

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.



Presented By:
Sheikh Muhammad Aamir

Click to add notes

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;
};
```

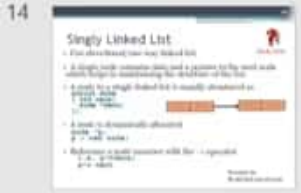
- A node is dynamically allocated

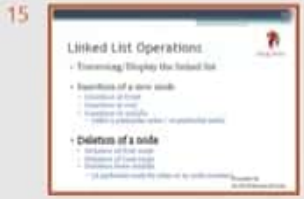
```
Node *p;
p = new Node;
```

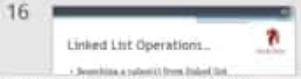
- Reference a node member with the -> operator

```
i.e. p->data;
p-> Next
```

Presented By:
Sheikh Muhammad Aamir







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

Linked List Operations...

Study Point

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

Presented By:
Sheikh Muhammad Aamir

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
```

Presented By:
Sheikh Muhammad Aamir

Click to add notes

Singly Linked List Program Understanding

Study Point

```
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
      .
      .
```

Presented By:
Sheikh Muhammad Aamir

Click to add notes

```
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

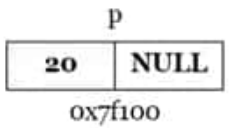
Presented By:
Sheikh Muhammad Aamir

Click to add notes

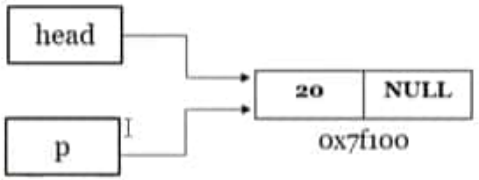
```
head = NULL;
createnode( );    //first call
```

Study Point

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



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



head = 0x7f100
p = 0x7f100

Presented By:
Sheikh Muhammad Aamir

Click to add notes

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

Study Point

p

30	NULL

0x7f200

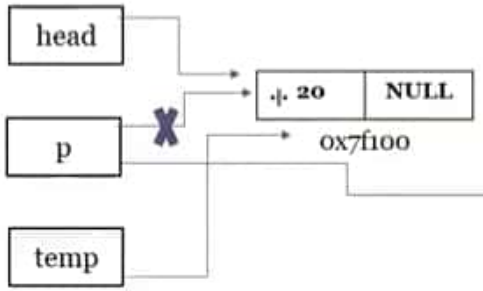
```
p = 0x7f200
p->data = 30
P->next = NULL
```

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

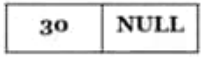
```
else
    { temp = head;
      while(temp->next != NULL)
     { temp = temp->next;
      }
     temp->next = p;
   }
```

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

Presented By:
Sheikh Muhammad Aamir

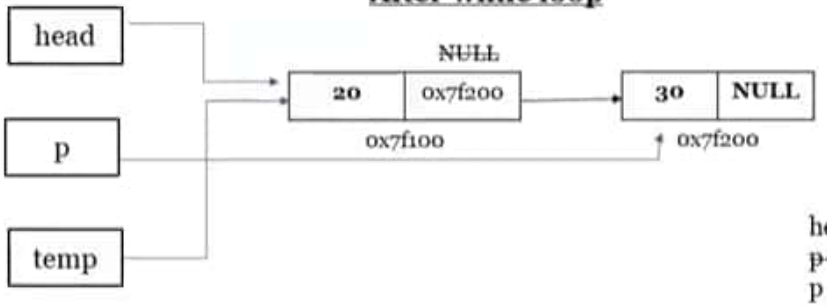


New Node Creation



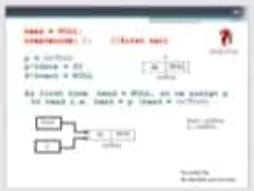
Study Point

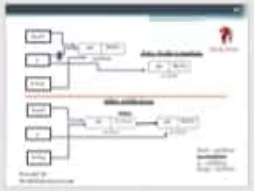
After while loop



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

Presented By:
Sheikh Muhammad Aamir

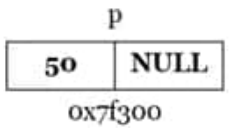





```
head = 0x7f100;
createnode( );      //third call
```

Study Point

```
p = 0x7f300
p->data = 50
P->next = NULL
```



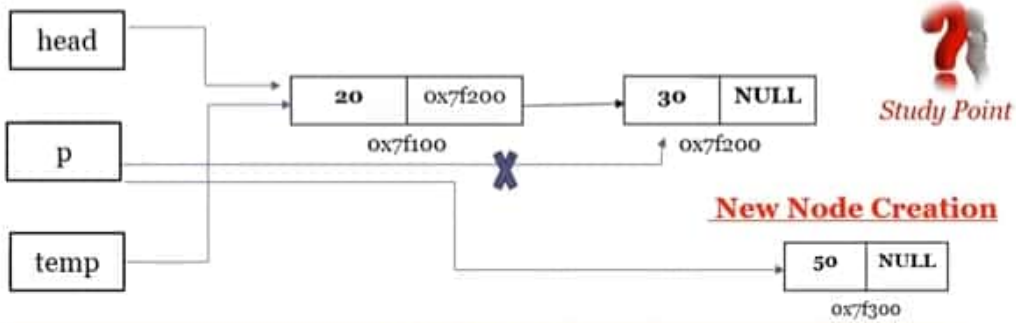
As head != NULL, so we will adopt else part

```
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

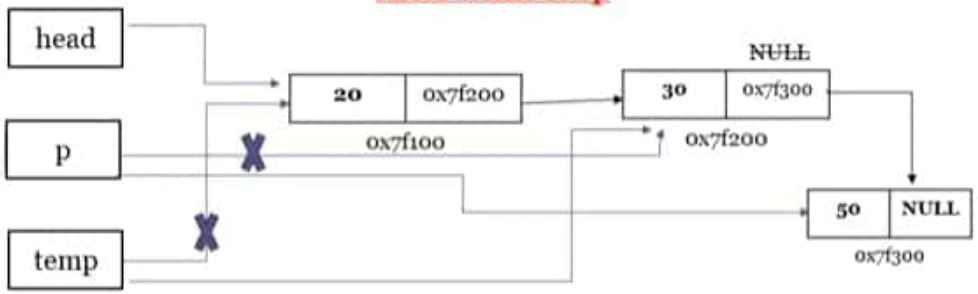
Click to add notes



Study Point

New Node Creation

After while loop



Presented By:
Sheikh Muhammad Aamir

head = 0xf100
p = 0x7f300
temp = ~~0x7f100, 0x7f200,~~ 0x7f300

```
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";
}
```

Study Point

Presented By:
Sheikh Muhammad Aamir

Click to add notes