06-02-2019

```
struct student {
    int roll_no;
    string name;
};

int main() {
    student s1;
    s1.name = "Ali";
    student *s;
    s = &s1;
    cout << (*s).name;
    cout << s->name;
```

student value

s1.name = "Ali" ✗

int x;
int *p; p = &x;

// one way of getting the address ...

// Ali

(*s).name
s1.name

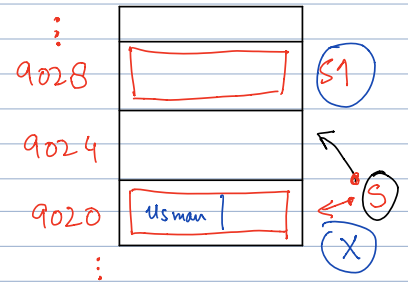
s-> ≡ (*s).

```
    // new instance
    s = new student;
    s->name = "Usman";
```

9020 ← pointer

new returns an address

(*s).name



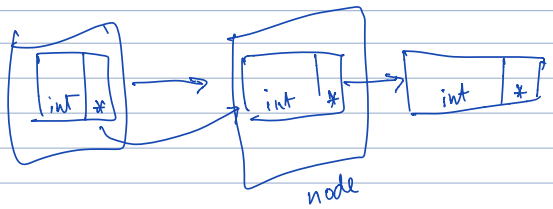

```
// delete an existing instance
//opposite of new
delete s;
```

Pointers Case Study

Recall Python's Lists : [1, 2, 3, 14, 16, 19]

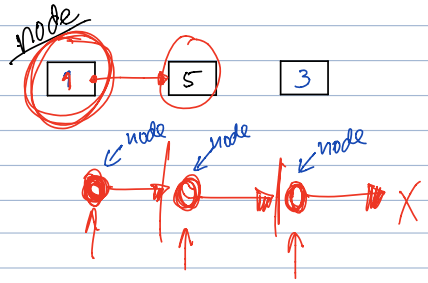
Important properties : (a) heterogenity ← structs
(b) variable length ←

Strategy : int a[1000000]; ✓ a[] = {1,2,3}; ✓ int a[]; ✗



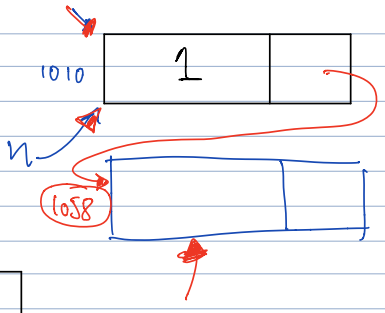
—

[1, 2, 3]




```
struct node {
    int val;
    node *next;
};
```

pointer




```
node *n;
n = NULL;
n = new node;
```

node*

address of a node

```
n -> val = 1;
n -> next = NULL;
```

```
n->next = new node;
```

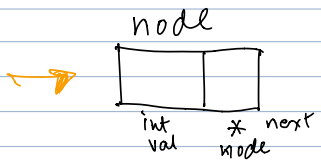
address of a node

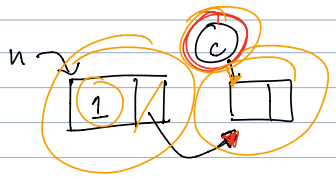
```
n->next = address1058;
```

```
cout << n->next;
```

```
n->next->val = 5;
(1058->val = 5;)
```

```
n->next->next = NULL;
```

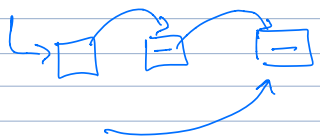


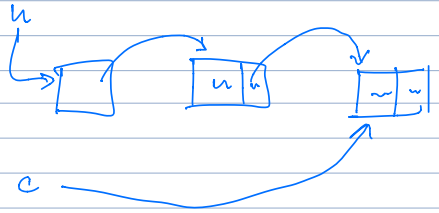


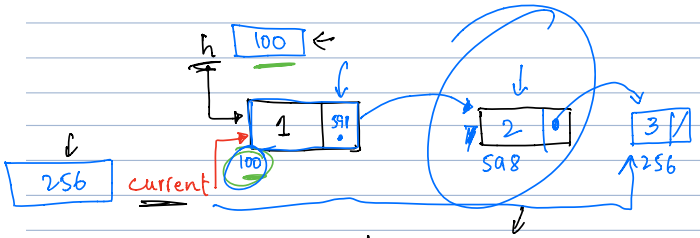

```
node *n;   node *current;
```

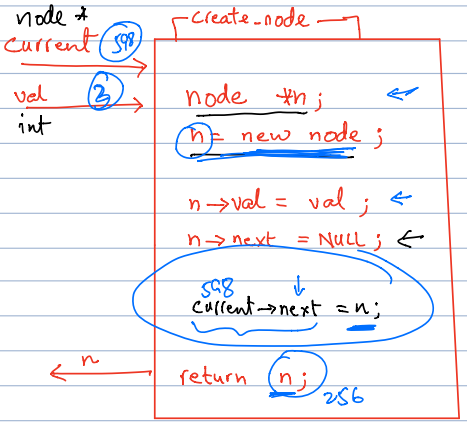
```
n = new node;
n -> val = 1;
n -> next = NULL;
```

```
current = new node;
n -> next = current;
current -> val = 5;
current -> next = NULL;
```









→ node *h; h = new node;

→ h → val = 1;

→ h → next = NULL;

→ node *current;

→ current = h;
100 100

→ current = create_node (current, 2);
598
100

current = create_node (current, 3);
598

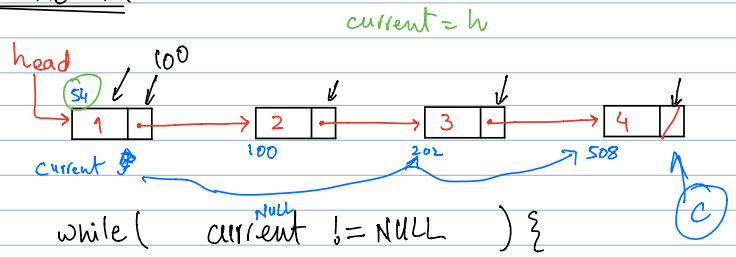
256

for (int i = 0; i < 10; i++) {

current = create_node (current, i);

}

Show the list:



while (current != NULL) {

cout << current → val << endl;

current = current → next;

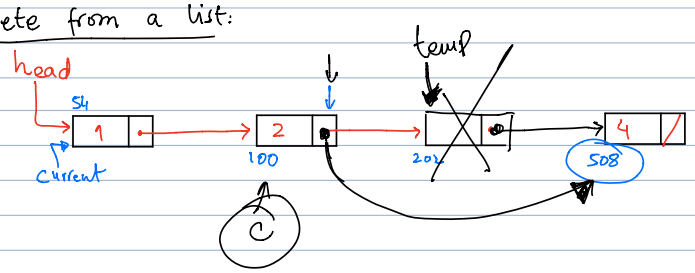
NULL NULL

}

current = h;
54

1
2
3
4

Delete from a list:



node * temp; temp = current -> next;

current -> next = current -> next -> next;

delete temp;

temp = NULL;