Linked Lists

Linked List data structures be like:
I know a guy who knows a guy

Advance Batch
– 22nd May

Agenda:

- Classes & Objects
- Why Linked Lists ?
- Basics

Classes & Objects

Real world problems

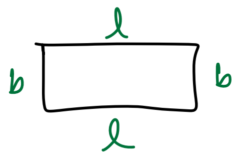
How to map real world things into our code ?

OOPS - Object Oriented Programming Style

→ Classes - Blueprint

→ Objects - Instance of real world object

Rectangle



Properties

→ Length

→ Breadth

Functionalities

→ Area

→ Perimeter

```
class    Rectangle   {

         int   length  = 0

         int   breadth = 0

}
```

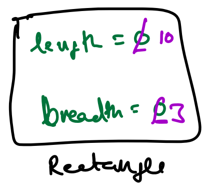
Blueprint for a rectangle which has 2 properties

```
r1 = new Rectangle ( )
```

→ An object of class Rectangle referred as r1

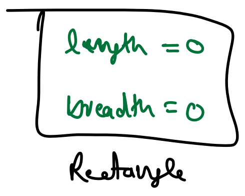
```
print (r1. length) → 0

print ( r1. breadth) → 0

r1 . length   =  10

r1 . breadth  =  3
```

r1 →

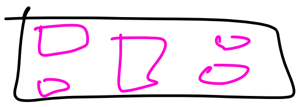


```
r2 = new Rectangle ( )

print( r2. breadth) → 0
```

r2 →



Design for
house



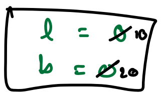
→ 10 storey building
10 houses

DNA → Humans

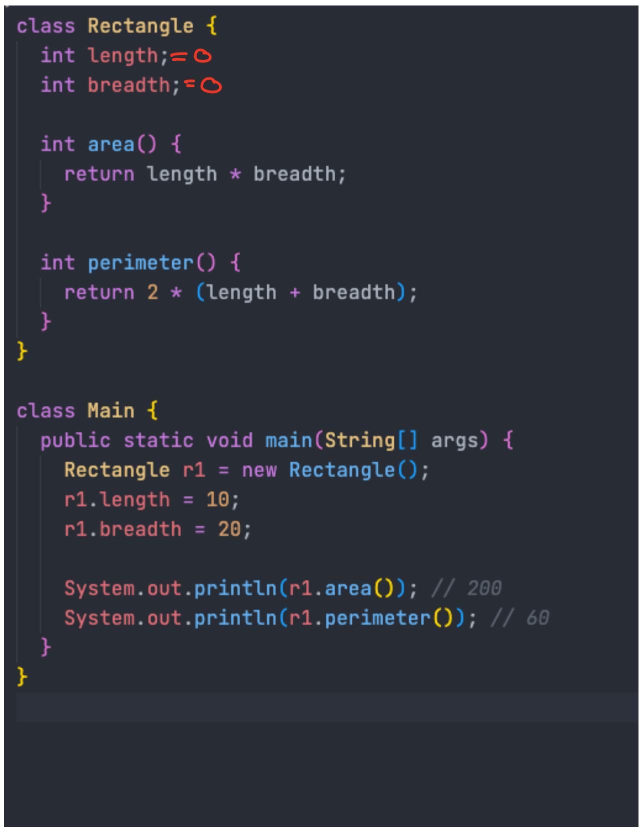
```
Class Rectangle {
    int length = 0
    int breadth = 0

    int area() {
        return length * breadth;
    }

    int perimeter() {
        return 2 (length + breadth)
    }
}
```

```
r1 = new Rectangle()
r1 . length = 10
```

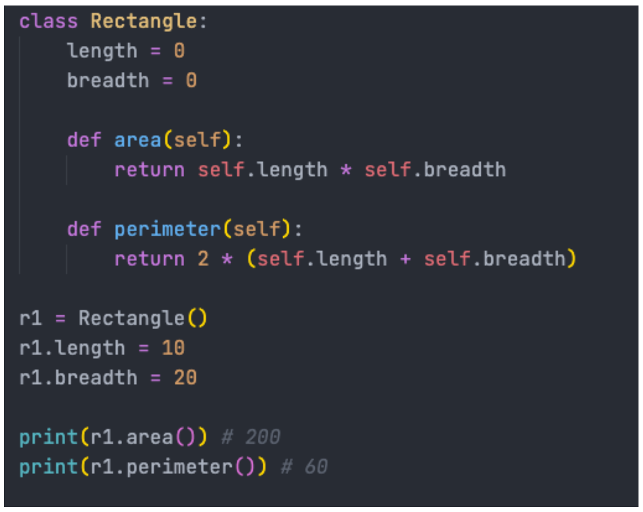
r1 →



```
r1. breadth = 20

print ( r1. area() )          → 200
print ( r1. perimeter () )    → 60


r2 = new Rectangle ()

r2. length = 40
r2. breadth = 100
```



```java
class Rectangle {
  int length;= 0
  int breadth;= 0

  int area() {
    return length * breadth;
  }

  int perimeter() {
    return 2 * (length + breadth);
  }
}

class Main {
  public static void main(String[] args) {
    Rectangle r1 = new Rectangle();
    r1.length = 10;
    r1.breadth = 20;

    System.out.println(r1.area()); // 200
    System.out.println(r1.perimeter()); // 60
  }
}
```



```python
class Rectangle:
    length = 0
    breadth = 0

    def area(self):
        return self.length * self.breadth

    def perimeter(self):
        return 2 * (self.length + self.breadth)

r1 = Rectangle()
r1.length = 10
r1.breadth = 20

print(r1.area()) # 200
print(r1.perimeter()) # 60
```

self – calling object

Constructors

1. A Constructor is special method that is called when an object is instantiated.
2. It must not return anything.
3. It is invoked automatically at the time of object construction.
4. It is generally used to initialise object properties

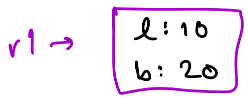
The name of the constructor is the same as that of class.

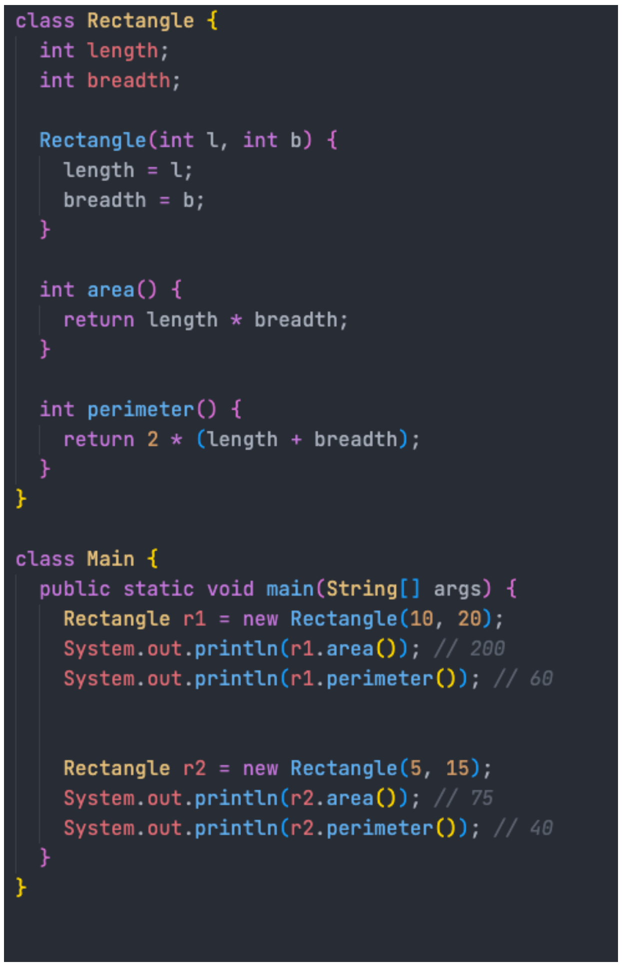
The name of the constructor is __init__()

```
class Rectangle {
        length = 0
        breadth = 0

        Rectangle ( l, b) {
                length = l
                breadth = b
        }
}
```

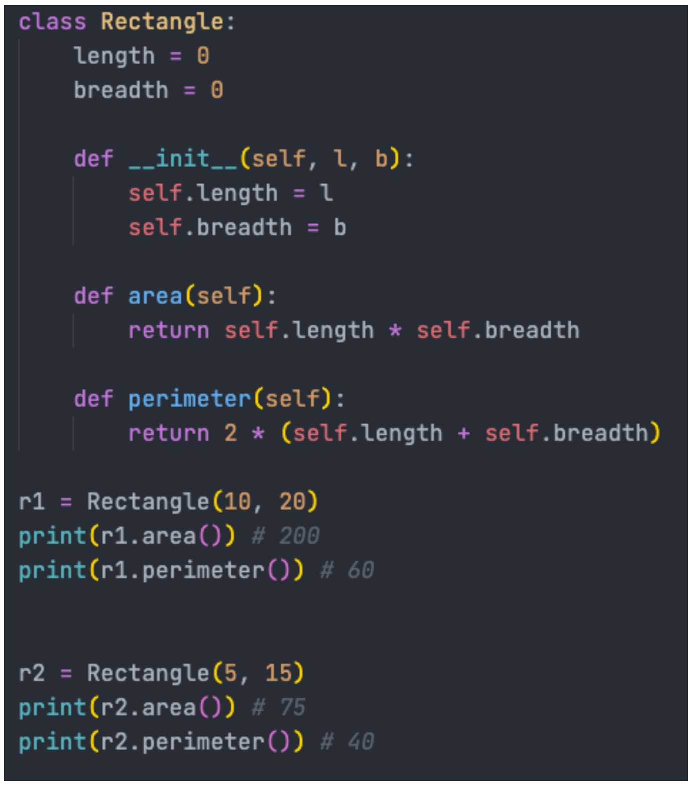
```
class Rectangle {
        length = 0
        breadth = 0

        def __init__ (self, l, b ) {
                length = l
                breadth = b
        }
}
```

↙ These args are passed to the constructor r1 = new Rectangle(10, 20)

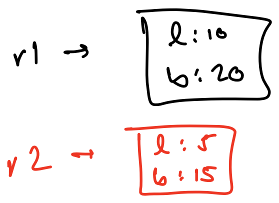




```java
class Rectangle {
  int length;
  int breadth;

  Rectangle(int l, int b) {
    length = l;
    breadth = b;
  }

  int area() {
    return length * breadth;
  }

  int perimeter() {
    return 2 * (length + breadth);
  }
}

class Main {
  public static void main(String[] args) {
    Rectangle r1 = new Rectangle(10, 20);
    System.out.println(r1.area()); // 200
    System.out.println(r1.perimeter()); // 60


    Rectangle r2 = new Rectangle(5, 15);
    System.out.println(r2.area()); // 75
    System.out.println(r2.perimeter()); // 40
  }
}
```



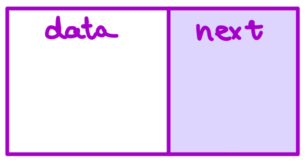
```python
class Rectangle:
    length = 0
    breadth = 0

    def __init__(self, l, b):
        self.length = l
        self.breadth = b

    def area(self):
        return self.length * self.breadth

    def perimeter(self):
        return 2 * (self.length + self.breadth)

r1 = Rectangle(10, 20)
print(r1.area()) # 200
print(r1.perimeter()) # 60


r2 = Rectangle(5, 15)
print(r2.area()) # 75
print(r2.perimeter()) # 40
```



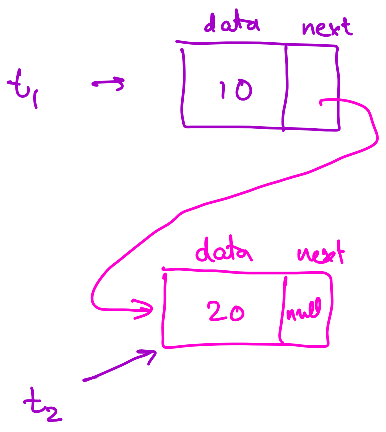
Break till 10:05 PM

Object Reference as Data Member

```
class Node {
    int data
    Node next

    constructor( x ) {
        data = x
        next = null
    }
}
```



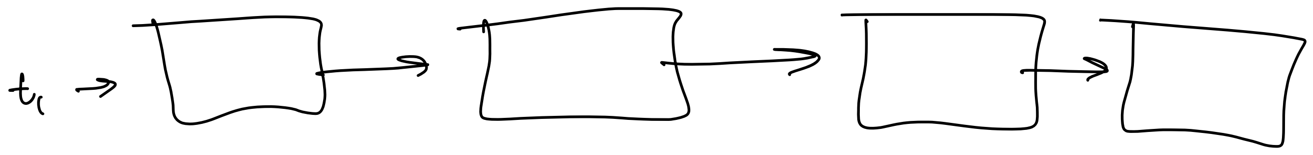
```
t1 = new Node(10)

t1.next = new Node (20)
```



$t_1.data \rightarrow 10$

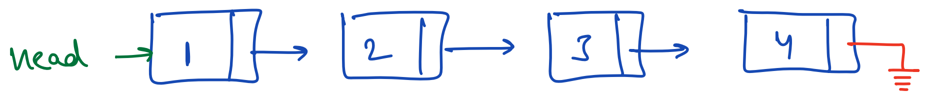
$t_1.next.data \rightarrow 20$

$t_2 = t_1.next$


t₁

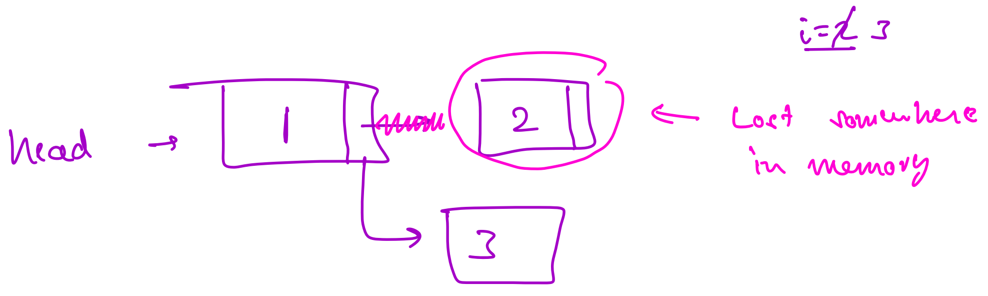
Q1 Given N > 0, create a linked list which contains data from 1 to N.

Example N = 4

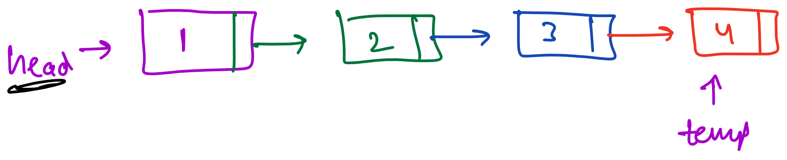


```
✓ head = new Node(1)

  for(i=2 ; i<=n ; i++) {

        head.next = new Node(i)

  }
```

N=4



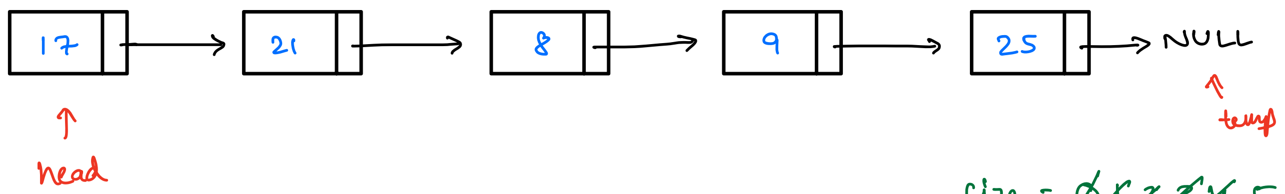
```
head = new Node(1)

temp = head

for( i = 2;   i <= n ; i++) {

        temp.next  =  new  Node(i)
        temp =  temp.next
}

return  head
```



Q2 Given head Node of a linked list, return its size.

Example

Ans = 5



size = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

Example

head → NULL

ans = 0

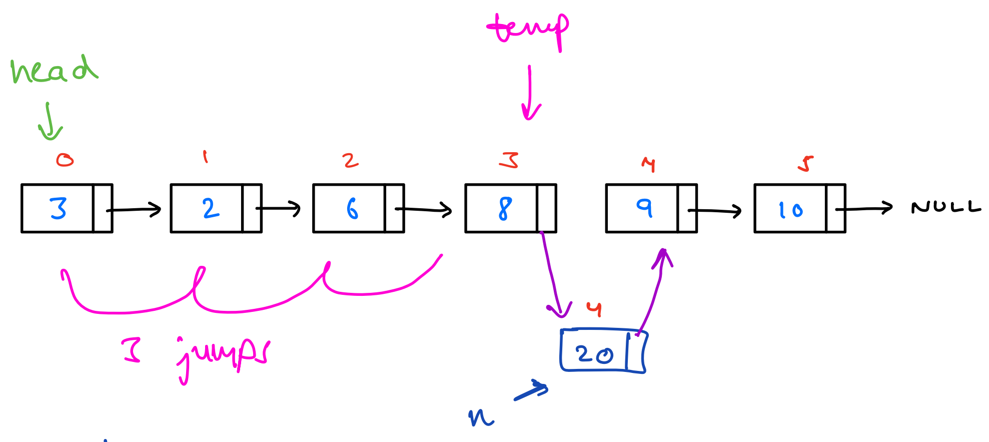
```
temp = head
size = 0
while ( temp ! = NULL) {
        size = size + 1

        temp = temp. next
}
return size
```

Q3 Given a head node of a LinkedList, insert a new node at kth index (0 based indexing)

Example 1

x = 20

k = 4



n.next = temp.next

temp . next = n

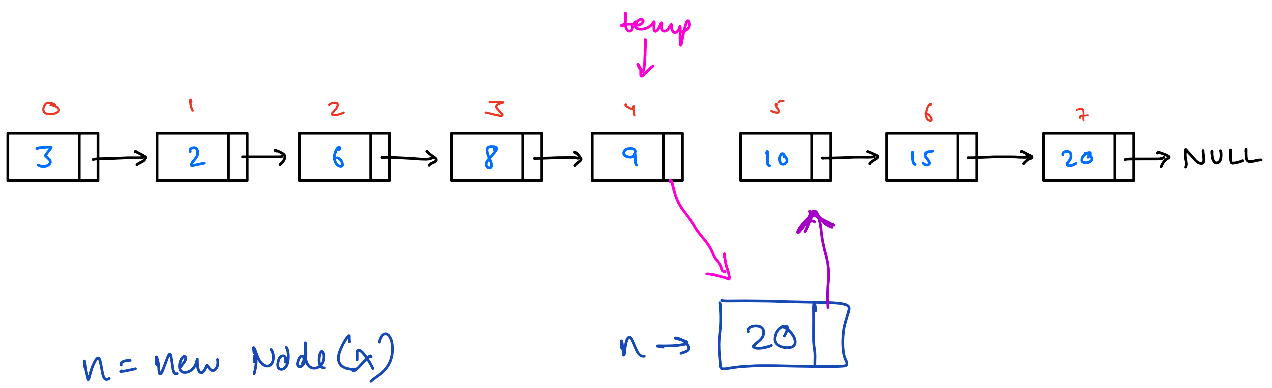
n= new Node (x)

Example 2

x = 20

k = 5

4 jumps



n= new Node (x)

n. next = temp.next

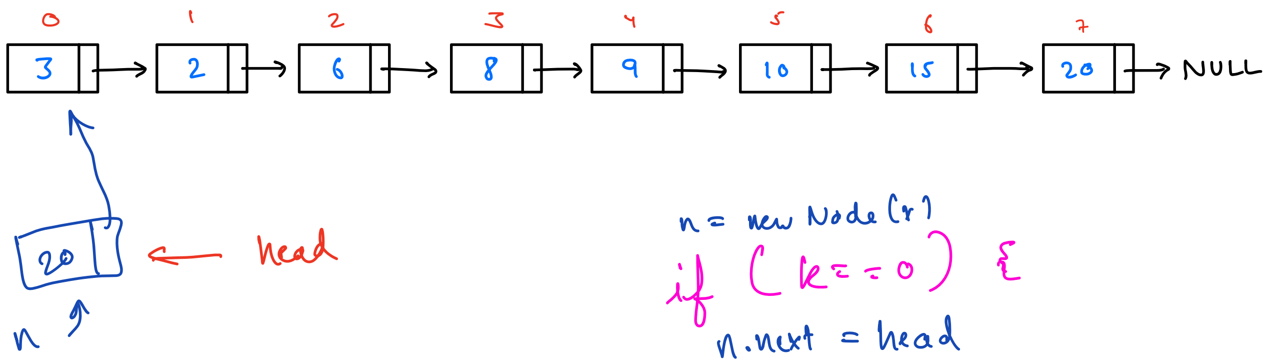
temp. next = n

Example 3

x = 20

k = 0

Corner case



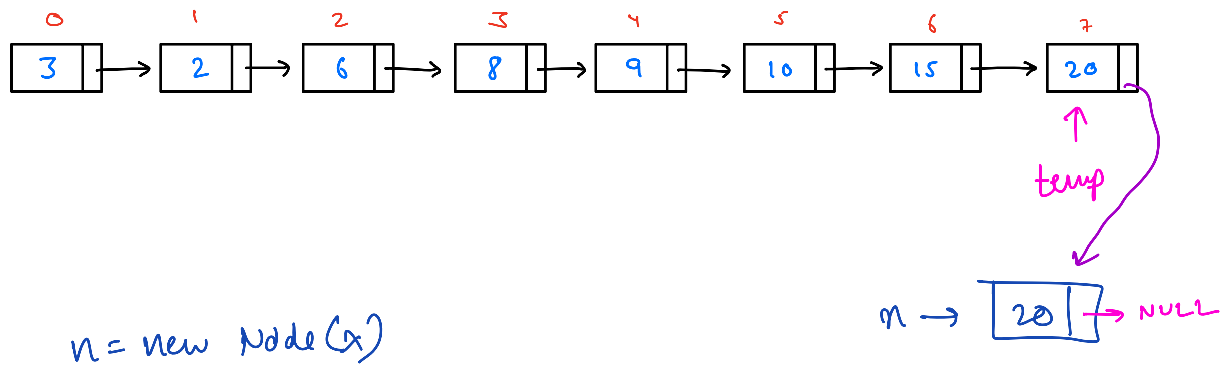
```
n = new Node(x)
if (k == 0) {
    n.next = head
    head = n
}
```

Example 4

7 jumps

$x = 20$

$k = 8$

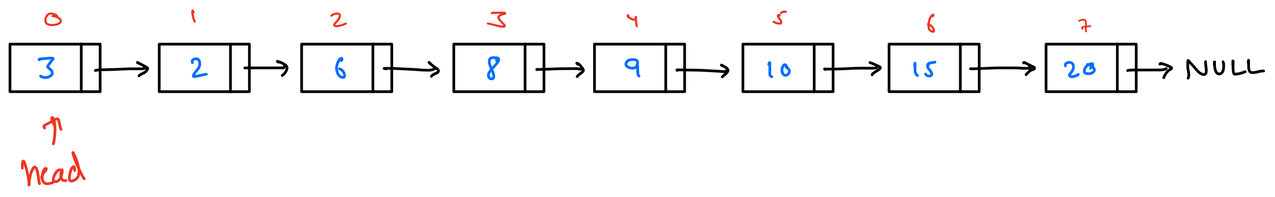


```
n = new Node(x)
n.next = temp.next
temp.next = n
```

Example 5

x = 20

k = 9



if (k > size of linked list)
    error / invalid input

Pseudocode

```
insert ( Node head, int k, int x ) {
        temp = head
        n = new Node(x)
        if ( k==0 ) {
            n.next = head
            head = n
            return head
        }
```

Previous question

```
    if( k  >  size (head)){

          print(" Invalid  input")

    }

    // k-1   jumps

    for (i=1;  i<= k-1 ;  i++) {

          temp=  temp.next

     }

    n. next  = temp.next

    temp. next  =  n

    return   head

}
```

Slides =

https://slides.com/tarunluthra/linked-lists-basics

Doubts

Python - Recursion

```
import sys
sys.setrecursionlimit(10**6)
```

} Recursion max depth reached error

Note: Do NOT use inbuilt Linked list in any language

Benefits

→ Arrays run out of memory

10^{10} entries

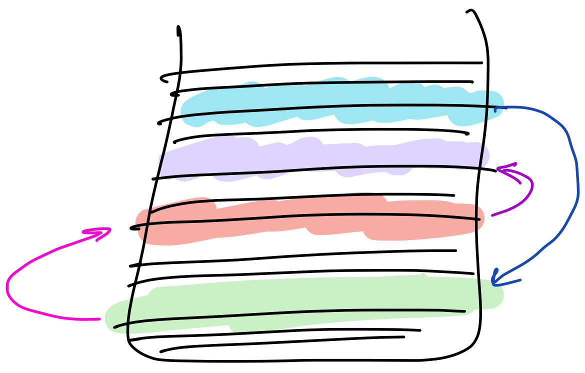
Write the

Node class

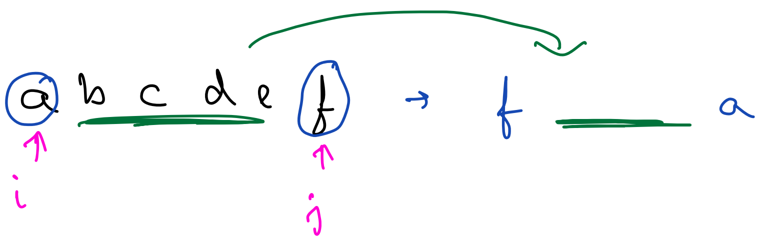
in every

Asn / HW question

(unless specified otherwise)

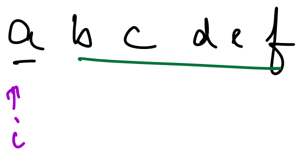


```
string    reverse ( string s, int i, int j) {
    if ( i > j)
            return s

    return  s[j]  +  reverse (s, i+1, j-1)    +  s[i]

}
```



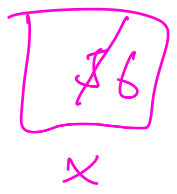
```
string reverse (string s, int i) {
    if ( i >= s.length)
        return "" ;

    return reverse(s, i+1) + s[i]
}
```

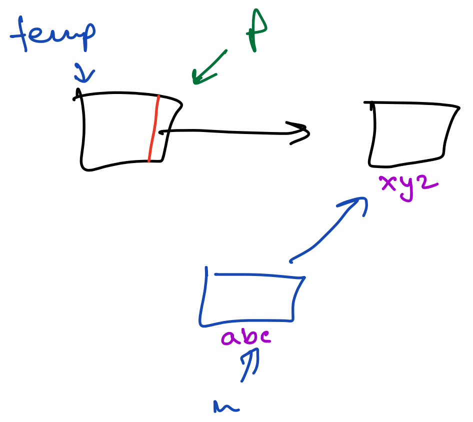
i=0

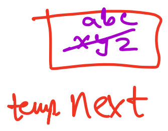


x=5

x=6







```
n.next = temp.next
temp.next = n
```

Instance Variable = Properties = Attributes

```
class Reetangle {
        length

        breadth
}
```

```
string reverse (string s, int i) {
    if ( i >= s.length)
        return " " ;

    return reverse(s, i+1) + s[i]
}
```

```
"cba"
reveve ("abc", 0) {
                 "cb"
    return  reveve ( "abc", 1) + "a"
}
                  reverse ("abc", 1) {
                                  "c"
                  return  reverse ("abc", 2) + "b"
                  }
```

```
            reverse ("abc", 2) {
                                      ..
            return   reverse("abc", 3) + "c"

        }
reverse("abc", 3) {
                 ..
    return
}
```

Good Night

Thank you

Wednesday

Last session