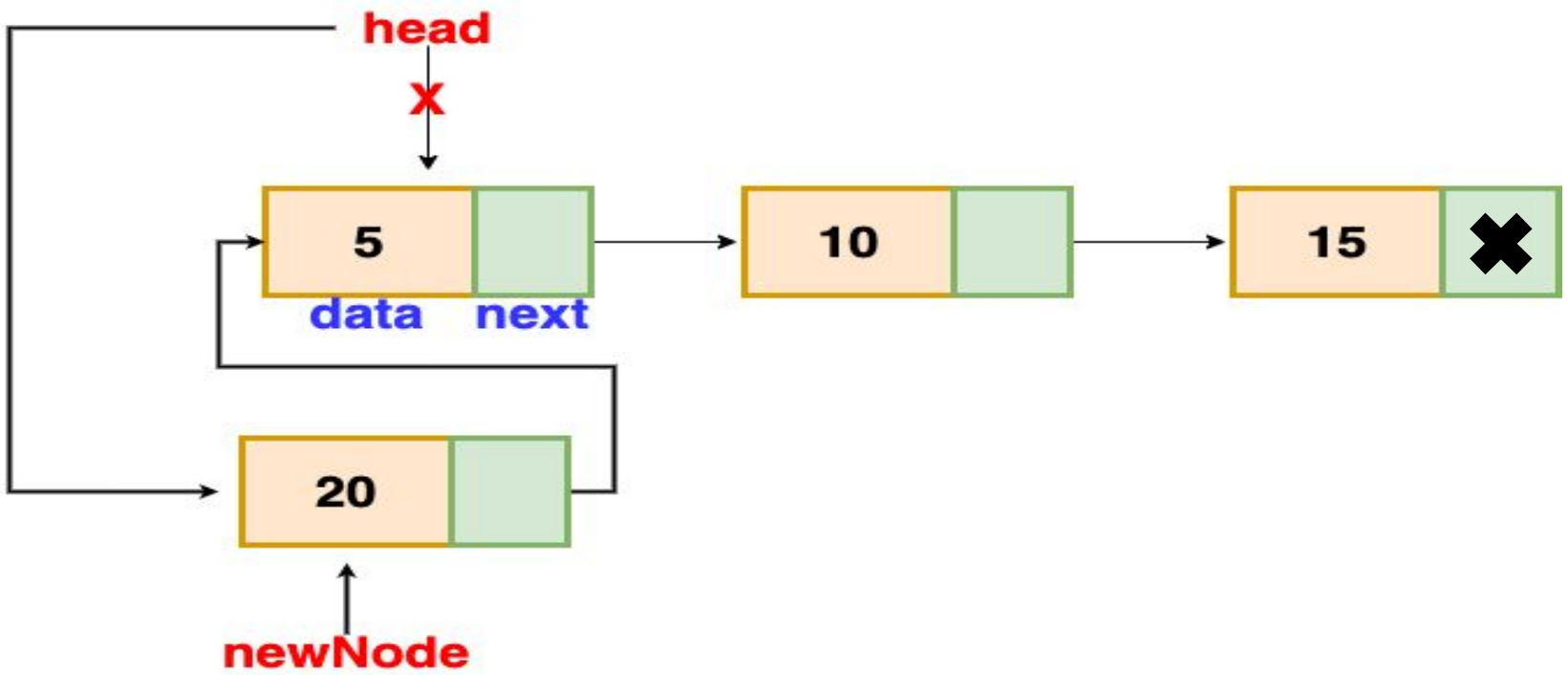
Insertion in a Single Linked List

- There are three possible positions where we can enter a new node in a linked list –
  - **Insertion at beginning**
  - **Insertion at end**
  - **Insertion at given position**

- Adding a new node in linked list is a more than one step activity.

Insertion in a Single Linked List (at beginning)

- Insertion at beginning



Insertion at the beginning

Insertion in single linked list (at beginning)

```
# A single node of a singly linked list
class Node:
def __init__(self, data):
    self.data = data
    self.next = None


# A Linked List class with a single
    head node
class LinkedList:
  def __init__(self):
    self.head = None
```

```
# insertion method for the linked list at
    beginning


def insert_beg(self, data):
    newNode = Node(data)
    if(self.head):
      newNode.next=self.head
      self.head=newNode
    else:
      self.head = newNode
```

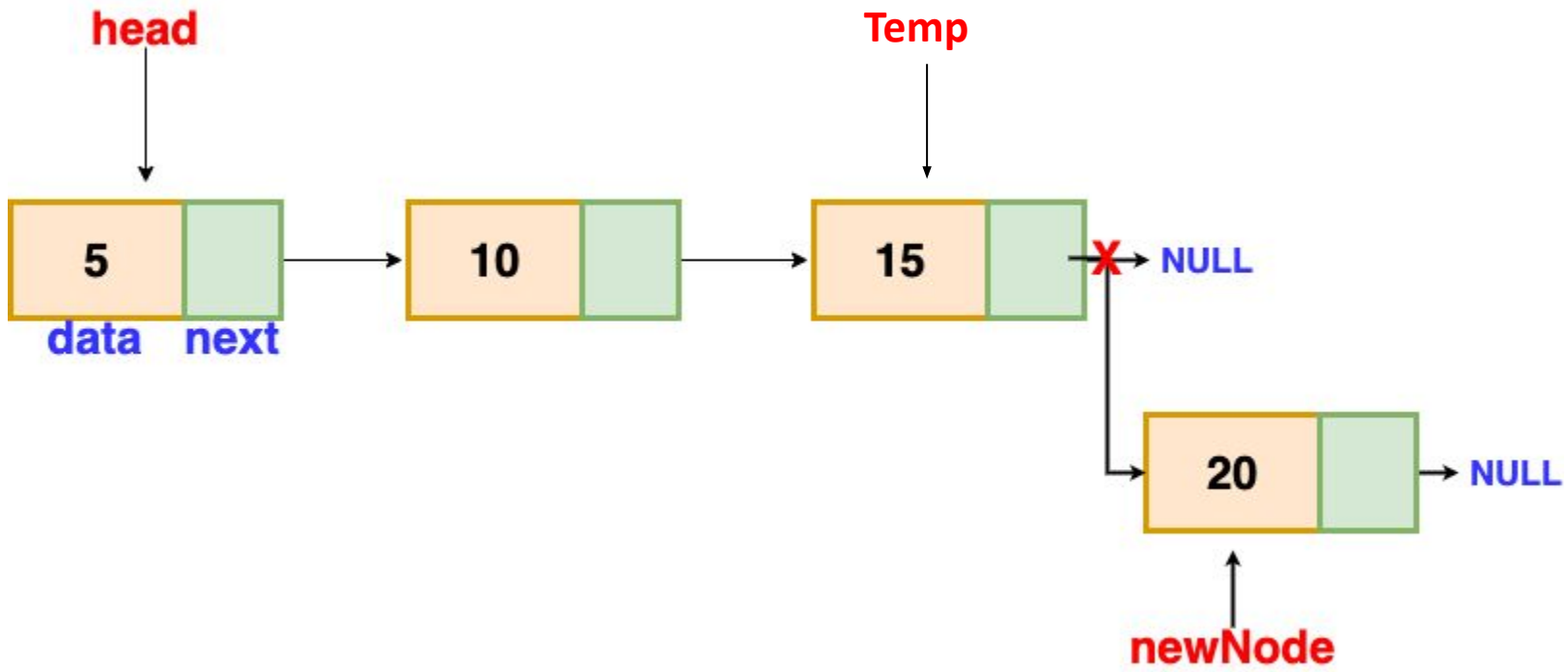
Insertion in single linked list (at beginning) (contd..)

```
# print method for the linked list
  def printLL(self):
    current = self.head
    if(current!=None):
        print("The List Contains:",end="\n")
        while(current):
            print(current.data)
            current = current.next
    else:
        print("List is Empty.")

# Singly Linked List with insertion and print methods
LL = LinkedList()
LL.insert_beg(3)
LL.insert_beg(4)
LL.insert_beg(5)
LL.printLL()
```

Insertion in a Single Linked List (at end)

- Insertion at end



Insertion at the end

Insertion in single linked list (at end)

```
# A single node of a singly linked list
class Node:
def __init__(self, data):
    self.data = data
    self.next = None


# A Linked List class with a single
    head node
class LinkedList:
  def __init__(self):
    self.head = None
```

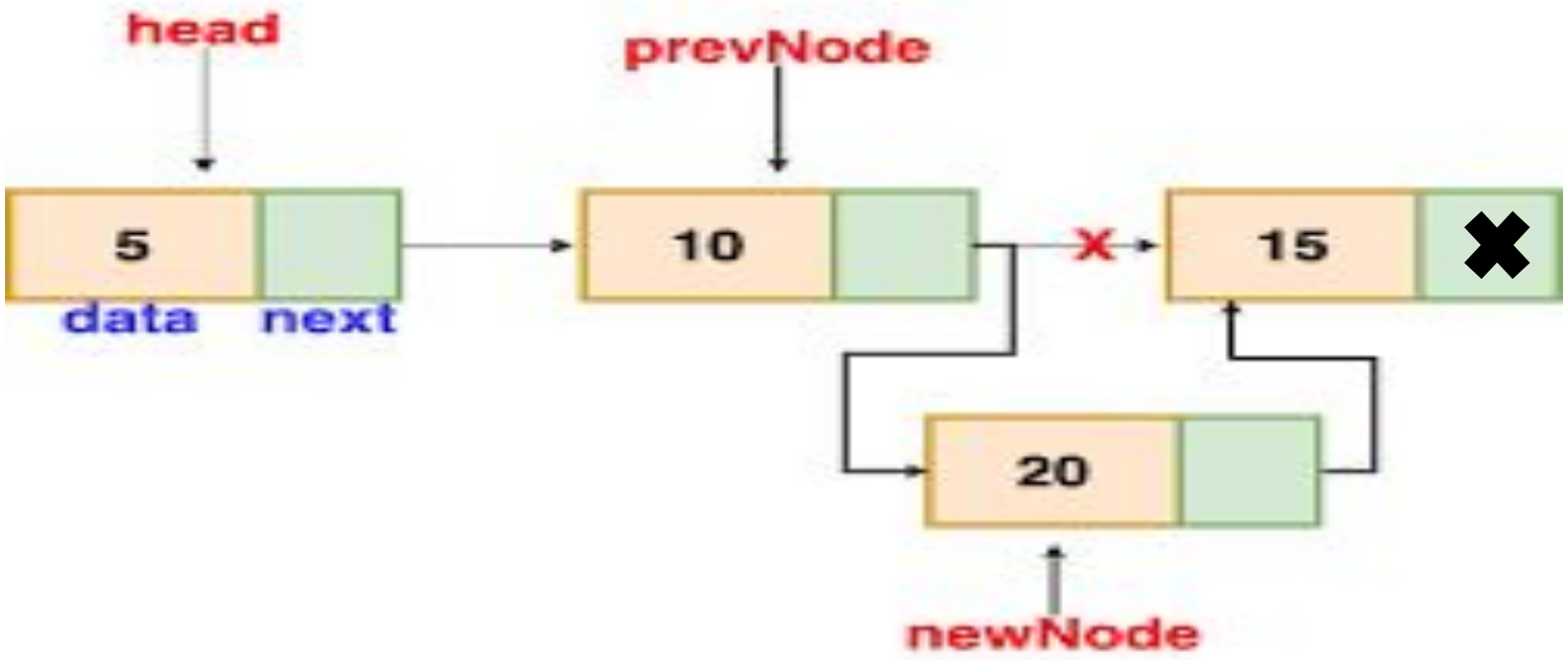
```
# insertion method for the linked list
     at end
  def insert_end(self, data):
   newNode = Node(data)
   if self.head  is None:
      self.head = newNode
   else:
    current = self.head
    while(current.next is not None):
      current = current.next
    current.next = newNode
```

Insertion in single linked list (at end) (contd..)

```
# print method for the linked list
  def printLL(self):
    current = self.head
    if(current!=None):
        print("The List Contains:",end="\n")
        while(current):
            print(current.data)
            current = current.next
    else:
        print("List is Empty.")
# Singly Linked List with insertion and print methods
LL = LinkedList()
LL.insert_end(3)
LL.insert_end(4)
LL.insert_end(5)
LL.printLL()
```

Insertion in a Single Linked List (at given position)

- Insertion at given position



Insertion after a given node

Insertion in single linked list (at position)

```
# A single node of a singly linked list
class Node:
def __init__(self, data):
    self.data = data
    self.next = None


# A Linked List class with a single
    head node
class LinkedList:
  def __init__(self):
    self.head = None
```

```
# creation method for the linked list
  def create(self, data):
    newNode = Node(data)
    if(self.head):
      current = self.head
      while(current.next):
        current = current.next
      current.next = newNode
    else:
      self.head = newNode
```

Insertion in single linked list (at position)

```
# insertion method for the linked list at
  given position


 def insert_position(self, data, pos):
  newNode = Node(data)
  if(pos<1):
     print("\nPosition should be >=1.")


  elif(pos==1):
     newNode.next=self.head
     self.head=newNode
else:
     current=self.head
     for i in range(1, pos-1):
        if(current!=None):
           current=current.next
     if(current!=None):
        newNode.next=current.next
        current.next=newNode
     else:
        print("\nThe previous node is null.")
```

Insertion in single linked list (at position)

```
# print method for the linked list
  def printLL(self):
    current = self.head
    if(current!=None):
        print("The List
    Contains:",end="\n")
        while(current):
            print(current.data)
            current = current.next
    else:
        print("List is Empty.")
```

```
# Singly Linked List with insertion and
    print methods
LL = LinkedList()
LL.create(2)
LL.create(3)
LL.create(4)
LL.create(5)
LL.create(6)
LL.insert_position(9, 4)
LL.printLL()
```