

# WALKING TECHIE

Blog About Java Programming, Design Pattern, And Data Structure.

[HOME](#) [JAVA](#) [DESIGN PATTERN](#) [SPRING BOOT](#) [SPRING BATCH](#)

## Categories

- > Array
- > Data Structure
- > Design Patterns
- > Git
- > Hibernate
- > Java
- > JPA
- > MongoDB
- > Spring Batch
- > Spring Boot
- > Spring Data
- > String

① x

[Home](#) x [Data Structure](#) x [Inserting a node in a singly linked list](#)

## Inserting a node in a singly linked list

👤 Santosh Kumar 📅 12/13/2018 04:41:00 PM 📁 Data Structure,

In the previous post, we have introduced about [Linked Lists](#). We have also discussed to [create a singly linked list with 3 nodes](#).

All program uses the below representation of the Singly linked list.

```
public class SinglyLinkedList<E> {  
  
    // instance variables of the SinglyLinkedList  
    private Node<E> head = null; // head node of the list (or null if  
  
    // ----- nested Node class -----  
    private static class Node<E> {  
        private E data;  
        private Node<E> next;  
  
        public Node(E data) {  
            this.data = data;  
        }  
    }  
    // ----- end of nested Node class -----  
}
```

In this post, we will discuss to insert a node in the linked list. Below are following ways to insert the node in the linked list.

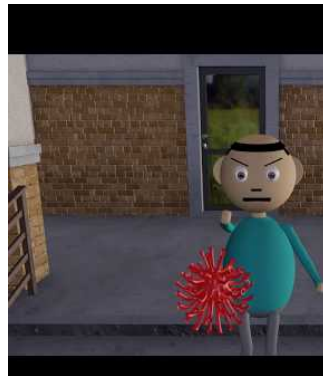
1. Insert a node at the head of a singly linked list.
2. Insert a node at the tail of a singly linked list.
3. Insert a node after given node of a singly linked list.

### Insert a node at the head of a singly linked list.

The new node is always inserted at front of the linked list and newly inserted node become the head of the linked list.

When using a singly linked list, we can easily insert an element at the head of the list. The main idea is that we create a new node, set its data(element) to the new node, set its next link to refer to the current head, and set the list's head to point to the new node.

Click below image to watch w  
comedy video



Enter your text...

RECENT POPULAR COM



git-log useful options

👤 Santosh Kumar 📅 Aug

How to see git configura

👤 Santosh Kumar 📅 Jul 2

How To Install and Config  
Ubuntu 18.04

👤 Santosh Kumar 📅 Jul 1

Difference between save  
persist method in Hiberr

👤 Santosh Kumar 📅 Jun

@AttributeOverride and  
@AttributeOverrides exa

👤 Santosh Kumar 📅 Jun

## Archive

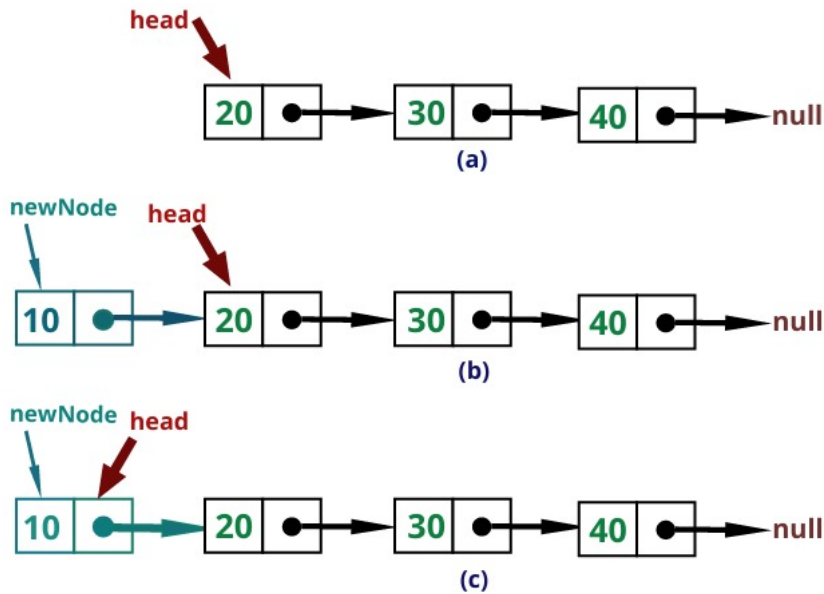
December ( 4 )

## Tags

ARRAY DATA STRUCTURE  
DESIGN PATTERNS GIT HIBER  
JAVA JPA MONGODB  
SPRING BATCH SPRING BOOT  
SPRING DATA STRING



Example: Let's take an example of singly linked list is **20->30->40** and we want to insert 10 at the head, then the linked list becomes **10->20->30->40**.



Let's implement a method call it as `addFirst` that will insert a node at the head of the linked list.



Following are the steps to add node at the head of the linked list.

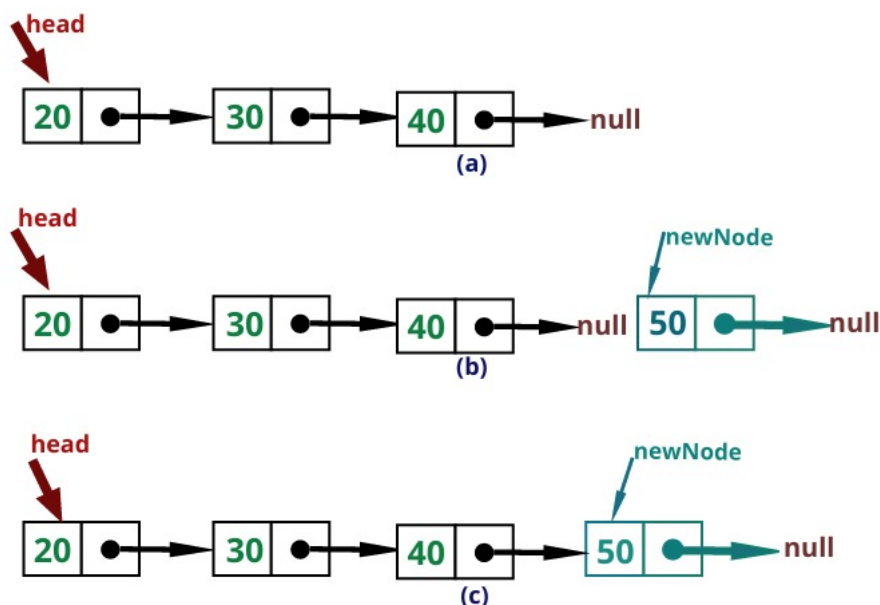
```
/* This method add a node at the head of the linked list or
 * you can say add a node at front of the linked list.*/
private void addFirst(E e) {
    // Create a new node and allocate element or data into it
    Node<E> newNode = new Node<E>(e);

    // refer next of new node to head
    newNode.next = this.head;

    // reassign head reference to the new node
    this.head = newNode;
}
```

### Insert a node at the tail of a singly linked list.

The new node is always added after the last node of the linked list. Let's take an example. If the given linked list is **20->30->40** and we want to insert 50 at the end of the linked list, then the linked list becomes **20->30->40->50**.



Here we are not maintaining the tail reference of the linked list so we need to traverse the linked list till end then change the last node's next to new node.

Following is the Java code to insert node at the end of the linked list.

```
/* This method add a new element at the tail of the linked list or
 * you can say at the end of the linked list.*/
private void addLast(E e) {
    // Create a new node and allocate element or data into it
    Node<E> newNode = new Node<E>(e);

    // New node is going to be the last node, so assign new's next as
    // refer next of new node to null
    newNode.next = null;

    // If the linked list is empty then make the new node as head and
    if (this.head == null) {
        newNode = this.head;
        return;
    }

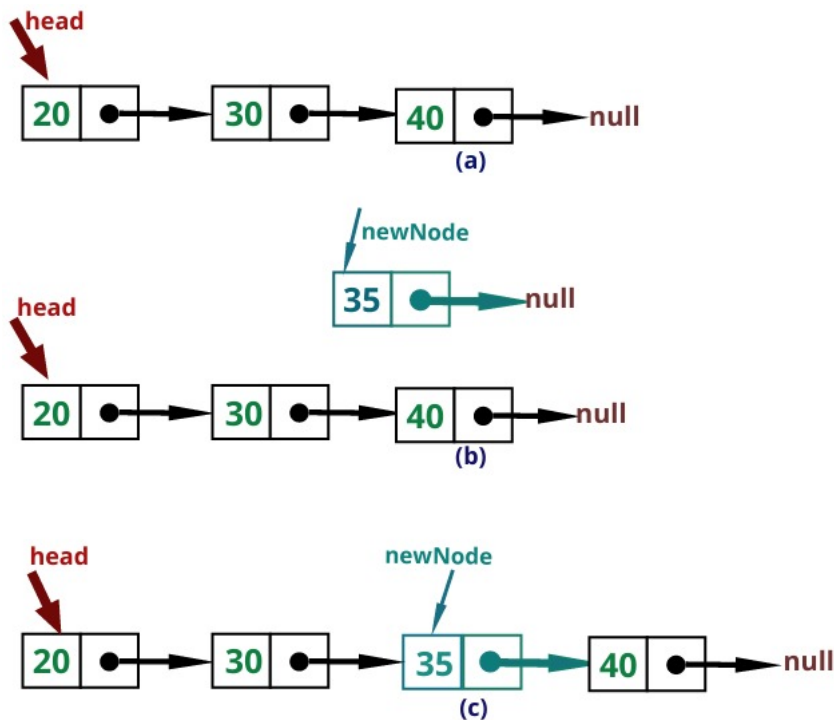
    // If linked list not empty then need to reach at the end of linked list
    Node<E> current = this.head;
    while (current.next != null) {
        current = current.next;
    }

    // reassign tail reference to the new node
    current.next = newNode;
}
```

### Insert a node after given node of a singly linked list.

To insert a node after the given node of the linked list, we will get the reference to the given node as the parameter in method call of `addAfter`.

Example: Let's take an example of singly linked list is **20->30->40** and we want to insert a node with data 35 after the given node 30 in the linked list, then the linked list becomes **20->30->35->40**.



Following is the Java code to insert a node after the given node. This method accept two parameter, one is an element and other is reference of the given node after which need to insert a node.

```
/*The addAfter method will add the new element or data after the giv
 * This method takes two parameters.*/
private void addAfter(E e, Node<E> givenNode) {
    // check if the given node is null and head is null
```

```

if (givenNode == null) {
    System.out.println("The given node can't be null.");
    return;
}

// check if the linked list is empty or head is null
if (this.head == null) {
    System.out.println("The given linked list is empty.");
    return;
}

// Create a new node and allocate element or data into it
Node<E> newNode = new Node<E>(e);

// assign newNode's next to givenNode's next
newNode.next = givenNode.next;

// assign givenNode's next as newNode
givenNode.next = newNode;
}

```

Following is the complete program of the insertion of node in the linked list. This program has consolidated all the discussed method above.

```

public class SinglyLinkedList<E> {

    // instance variables of the SinglyLinkedList
    private Node<E> head = null; // head node of the list (or null if

    // ----- nested Node class -----
    private static class Node<E> {
        private E data;
        private Node<E> next;

        public Node(E data) {
            this.data = data;
        }
    }
    // ----- end of nested Node class -----

    /* instance method of Singly linked list
    * which will print the data of nodes starting from the head.*/
    private void printList() {
        Node<E> current = this.head;

        while (current != null) {
            System.out.print(current.data + "->");
            current = current.next;
        }
        System.out.println("null");
    }

    // This method adds a node at the head of the linked list or
    // you can say add a node at front of the linked list.
    private void addFirst(E e) {
        // Create a new node and allocate element or data into it
        Node<E> newNode = new Node<E>(e);

        // refer next of new node to head
        newNode.next = this.head;

        // reassign head reference to the new node
        this.head = newNode;
    }

    // This method adds a new element at the tail of the linked list or
    // you can say at the end of the linked list.
    private void addLast(E e) {
        // Create a new node and allocate element or data into it
        Node<E> newNode = new Node<E>(e);

        // New node is going to be the last node, so assign new's next as
        // refer next of new node to null
        newNode.next = null;

        // If the linked list is empty then make the new node as head or
        if (this.head == null) {
            newNode = this.head;
            return;
        }

        // If linked list not empty then need to reach at the end of list
        Node<E> current = this.head;
        while (current.next != null) {

```

```

        current = current.next;
    }

    // reassign tail reference to the new node
    current.next = newNode;
}

// The addAfter method will add the new element or data after the
// This method takes two parameters.
private void addAfter(E e, Node<E> givenNode) {
    // check if the given node is null and head is null
    if (givenNode == null) {
        System.out.println("The given node can't be null.");
        return;
    }

    // check if the linked list is empty or head is null
    if (this.head == null) {
        System.out.println("The given linked list is empty.");
        return;
    }

    // Create a new node and allocate element or data into it
    Node<E> newNode = new Node<E>(e);

    // assign newNode's next to givenNode's next
    newNode.next = givenNode.next;

    // assign givenNode's next as newNode
    givenNode.next = newNode;
}

// method to create a simple linked list with 3 nodes
public static void main(String[] args) {
    // Start with the empty list.
    SinglyLinkedList<Integer> singlyLinkedList = new SinglyLinkedList();

    // add element to the front of the linked list
    // Linked list become 15->null
    singlyLinkedList.addFirst(15);

    // add element to the end of the linked list
    // Linked list become 15->50->null
    singlyLinkedList.addLast(50);

    // add element 18 after node's element is 15
    // linked list now become 15->18->50->null
    singlyLinkedList.addAfter(18, singlyLinkedList.head);

    // add element 55 after node's element is 18
    // linked list now become 15->18->55->50->null
    singlyLinkedList.addAfter(55, singlyLinkedList.head.next);

    // add element 10 to the front of the linked list
    // linked list now become 10->15->18->55->50->null
    singlyLinkedList.addFirst(10);

    // print the elements of the linked list
    singlyLinkedList.printList();
}
}

```

Output:

10->15->18->55->50->null

Tags # Data Structure

 TWEET

 SHARE

 PIN IT

 COMMENT

 Author Image

### About Santosh Kumar

Dreamer, passionate who loves Java and open stuff.

Inserting a node in a singly  
linked list

Implementation of singly  
linked list

Introduction to Linked List in  
Data Structure



### Data Science & PostgreSQL

From Machine Learning, A.I. and Big Data consulting and training


CYBERTEC

[Newer Post](#)[Older Post](#)

BLOGGER

NO COMMENTS :

POST A COMMENT

 Comment as: ahm7dkhalifa@ ▼ Sign out

Publish Preview ☐ Notify me

### Labels

ARRAY DATA STRUCTURE  
DESIGN PATTERNS GIT HIBERNATE  
JAVA JPA MONGODB SPRING BATCH  
SPRING BOOT SPRING DATA STRING

### Business

### Contact Form

Send