**Java program to delete a node from the middle of the singly linked list .**

**package** pblm;

**public** **class** pro2{

**class** Node{

**int** data;

Node next;

**public** Node(**int** data)

{

**this**.data = data;

**this**.next = **null**;

}

}

**public** Node head = **null**;

**public** Node tail = **null**;

**public** **int** size;

**public** **void** addNode(**int** data) {

Node newNode = **new** Node(data);

**if**(head == **null**) {

head = newNode;

tail = newNode;

}

**else** {

tail.next = newNode;

tail = newNode;

}

size++;

}

**void** deleteFromMid() {

Node temp, current;

**if**(head == **null**) {

System.***out***.println("List is empty");

**return**;

}

**else** {

**int** count = (size % 2 == 0) ? (size/2) : ((size+1)/2);

**if**( head != tail ) {

temp = head;

current = **null**;

**for**(**int** i = 0; i < count-1; i++){

current = temp;

temp = temp.next;

}

**if**(current != **null**) {

current.next = temp.next;

temp = **null**;

}

**else** {

head = tail = temp.next;

temp = **null**;

}

}

**else** {

head = tail = **null**;

}

}

size--;

}

**public** **void** display() {

Node current = head;

**if**(head == **null**) {

System.***out***.println("List is empty");

**return**;

}

**while**(current != **null**) {

System.***out***.print(current.data + " ");

current = current.next;

}

System.***out***.println();

}

**public** **static** **void** main(String[] args) {

pro2 sList = **new** pro2();

sList.addNode(1);

sList.addNode(2);

sList.addNode(3);

sList.addNode(4);

System.***out***.println("Original List: ");

sList.display();

**while**(sList.head != **null**) {

sList.deleteFromMid();

System.***out***.println("Updated List: ");

sList.display();

}

}

}

**Output:**

