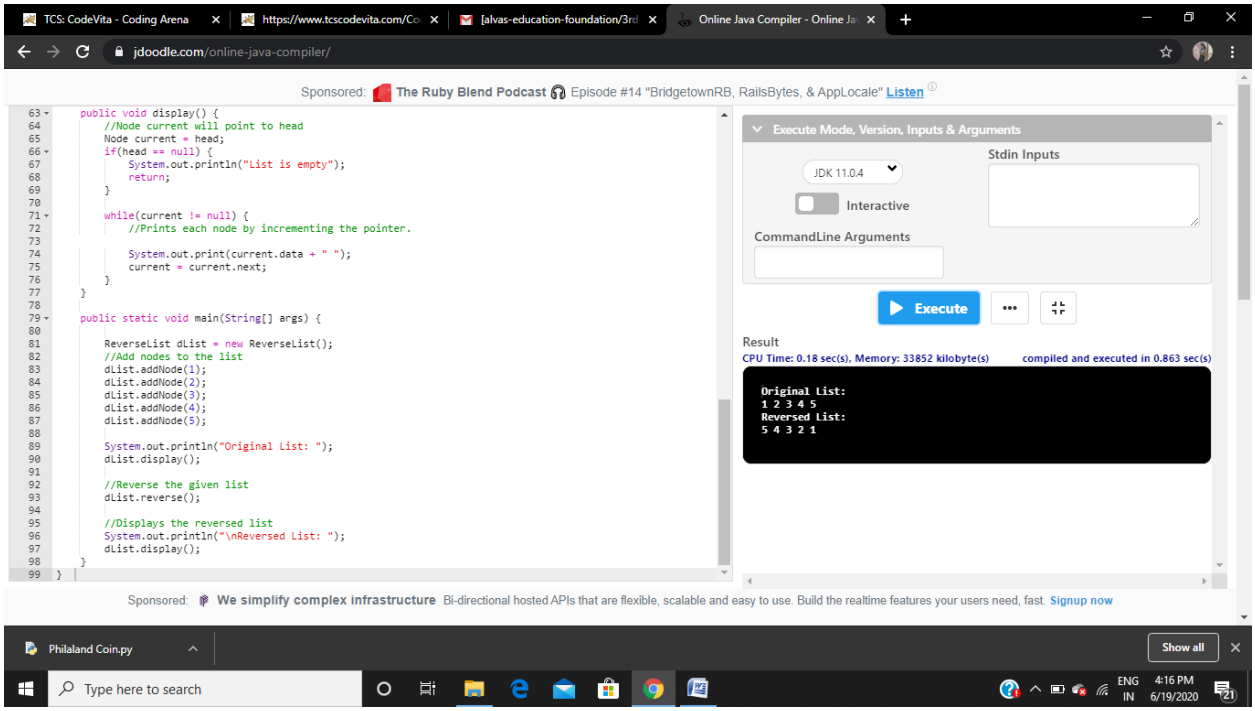
Write a Java program to create a doubly linked list of n nodes and display it in reverse order

```
public class ReverseList {


  //Represent a node of the doubly linked list


  class Node{
    int data;
    Node previous;
    Node next;


    public Node(int data) {
      this.data = data;
    }
  }


  //Represent the head and tail of the doubly linked list
  Node head, tail = null;


  //addNode() will add a node to the list
  public void addNode(int data) {
    //Create a new node
    Node newNode = new Node(data);


    //If list is empty
    if(head == null) {
```

```
        //Both head and tail will point to newNode
        head = tail = newNode;
        //head's previous will point to null
        head.previous = null;
        //tail's next will point to null, as it is the last node of the list
        tail.next = null;
    }
    else {
        //newNode will be added after tail such that tail's next will point to newNode
        tail.next = newNode;
        //newNode's previous will point to tail
        newNode.previous = tail;
        //newNode will become new tail
        tail = newNode;
        //As it is last node, tail's next will point to null
        tail.next = null;
    }
}

//reverse() will reverse the doubly linked list
public void reverse() {
    //Node current will point to head
    Node current = head, temp = null;

    //Swap the previous and next pointers of each node to reverse the direction of the list
```

```
    while(current != null) {
        temp = current.next;
        current.next = current.previous;
        current.previous = temp;
        current = current.previous;
    }
    //Swap the head and tail pointers.
    temp = head;
    head = tail;
    tail = temp;
}

//display() will print out the elements of the list
public void display() {
    //Node current will point to head
    Node current = head;
    if(head == null) {
        System.out.println("List is empty");
        return;
    }

    while(current != null) {
        //Prints each node by incrementing the pointer.

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

```
            current = current.next;
        }
    }

    public static void main(String[] args) {

        ReverseList dList = new ReverseList();
        //Add nodes to the list
        dList.addNode(1);
        dList.addNode(2);
        dList.addNode(3);
        dList.addNode(4);
        dList.addNode(5);

        System.out.println("Original List: ");
        dList.display();

        //Reverse the given list
        dList.reverse();

        //Displays the reversed list
        System.out.println("\nReversed List: ");
        dList.display();
    }
}
```


TCS: CodeVita - Coding Arena
https://www.tcscodevita.com/Co
[alvas-education-foundation/3rd
Online Java Compiler - Online Ja
jdoodle.com/online-java-compiler/
Sponsored: The Ruby Blend Podcast Episode #14 "BridgetownRB, RailsBytes, & AppLocale" Listen
public void display() {
//Node current will point to head
Node current = head;
if(head == null) {
System.out.println("List is empty");
return;
}
while(current != null) {
//Prints each node by incrementing the pointer.
System.out.print(current.data + " ");
current = current.next;
}
}
public static void main(String[] args) {
ReverseList dList = new ReverseList();
//Add nodes to the list
dList.addNode(1);
dList.addNode(2);
dList.addNode(3);
dList.addNode(4);
dList.addNode(5);
System.out.println("Original List: ");
dList.display();
//Reverse the given list
dList.reverse();
//Displays the reversed list
System.out.println("\nReversed List: ");
dList.display();
}
}
Execute Mode, Version, Inputs & Arguments
JDK 11.0.4
Interactive
Stdin Inputs
CommandLine Arguments
Execute
Result
CPU Time: 0.18 sec(s), Memory: 33852 kilobyte(s)
compiled and executed in 0.863 sec(s)
Original List:
1 2 3 4 5
Reversed List:
5 4 3 2 1
Sponsored: We simplify complex infrastructure Bi-directional hosted APIs that are flexible, scalable and easy to use. Build the realtime features your users need, fast. Signup now
Philaland Coin.py
Show all
Type here to search
ENG
IN
4:16 PM
6/19/2020