```java
package linkedlist;

// Java program to detect loop in a linked list
import java.util.*;

  class LinkedList4 {

    static Node head; // head of list

    /* Linked list Node*/
    static class Node {
        int data;
        Node next;
        Node(int d)
        {
            data = d;
            next = null;
        }
    }

    /* Inserts a new Node at front of the list. */
    static public void push(int new_data)
    {
        /* 1 & 2: Allocate the Node &
                  Put in the data*/
        Node new_node = new Node(new_data);

        /* 3. Make next of new Node as head */
        new_node.next = head;

        /* 4. Move the head to point to new Node */
        head = new_node;
    }

    // Returns true if there is a loop in linked
    // list else returns false.
    static boolean detectLoop(Node h)
    {
        HashSet<Node> s = new HashSet<Node>();
        while (h != null) {
            // If we have already has this node
            // in hashmap it means their is a cycle
            // (Because you we encountering the
            // node second time).
            if (s.contains(h))
                return true;

            // If we are seeing the node for
            // the first time, insert it in hash
            s.add(h);

            h = h.next;
        }

        return false;
    }

    /* Driver program to test above function */
    public static void main(String[] args)
    {
        LinkedList4 llist = new LinkedList4();

        llist.push(20);
        llist.push(4);
        llist.push(15);
        llist.push(10);
```

```
        /*Create loop for testing */
        llist.head.next.next.next.next = llist.head;

        if (detectLoop(head))
            System.out.println("Loop found");
        else
            System.out.println("No Loop");
    }
}

// This code is contributed by Arnav Kr. Mandal.
```