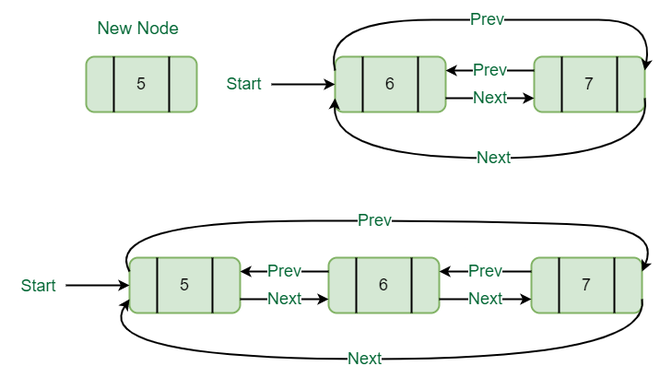
***Circular Doubly Linked List***

Circular Doubly Linked List has properties of both doubly linked list and circular linked list in which two consecutive elements are linked or connected by the previous and next pointer and the last node points to the first node by the next pointer and also the first node points to the last node by the previous pointer.

**Insertion at the beginning of the list:**

To insert a node at the beginning of the list, create a node(Say T) with data = 5, T next pointer points to the first node of the list, T previous pointer points to the last node of the list, last node’s next pointer points to this T node, first node’s previous pointer also points this T node and at last don’t forget to shift ‘Start’ pointer to this T node.



*Insertion at the beginning of the list*

Below is the implementation of the above operation:

C++Java

// Function to insert Node at the beginning

// of the List,

static void insertBegin(int value)

{

// Pointer points to last Node

Node last = (start).prev;

Node new\_node = new Node();

new\_node.data = value; // Inserting the data

// setting up previous and next of new node

new\_node.next = start;

new\_node.prev = last;

// Update next and previous pointers of start

// and last.

last.next = (start).prev = new\_node;

// Update start pointer

start = new\_node;

}