

PROJECT 1 - LINKED LISTS

CyclicLinkedList

```

- Singlenode< T > * p // head
- Singlenode< T > * q // tail
- int n // size
+ CyclicLinkedList()
: head(null), tail(null), n(0)
+ ~CyclicLinkedList()
+ int size(void) const
+ bool empty(void) const
+ ~T front(void) const
+ Singlenode< T > * head(void) const
+ Singlenode< T > * tail(void) const
+ int count(T const &) const
+ void push_front(T, const &)
+ void push_back(T, const &)
+ ~T pop_front(void)
+ int erase(T const &)

```

DoublyLinkedList

```

- Doublenode< T > * h // head
- Doublenode< T > * t // tail
- int n // size
+ DoublyLinkedList()
+ ~DoublyLinkedList()
+ int size(void) const
+ bool empty(void) const
+ ~T front(void) const
+ ~T back(void) const
+ Doublenode< T > * head(void) const
+ Doublenode< T > * tail(void) const
+ int count(T const &) const
+ void push_front(T, const &)
+ void push_back(T, const &)
+ ~T pop_front(void)
+ int erase(T const &)

```

SingleNode

```

Friend class CyclicLinkedList
- T data
- Singlenode * next
+ Singlenode(T type const &, Singlenode*)
+ ~T getData(void)
+ Singlenode* getNext(void) const
+ ~Singlenode()

```

DoubleNode

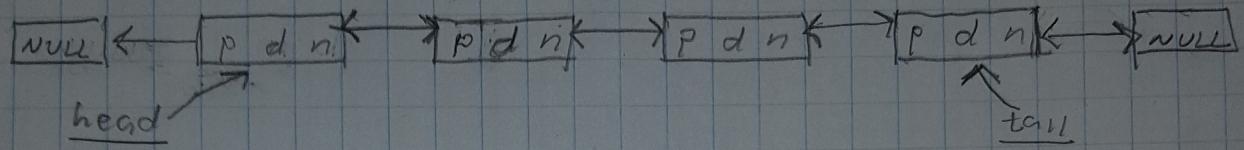
```

Friend class DoublyLinkedList
- T data
- Doublenode* next
- Doublenode* previous
+ Doublenode(T type const &, Doublenode*, Doublenode*)
+ ~Doublenode()
+ ~T getData(void) const
+ Doublenode* getNext(void) const
+ Doublenode* getPrevious(void) const

```

DoublyLinkedList

p ; = previous , d ; = data , n ; = next



CyclicLinkedList

d ; = data , n ; = next

