CS235102
Data Structure
Homework 2

2017/10/24 10:00 ~

2017/11/07 23:59

(Hard Deadline)

Target

- The target of this homework is to implement a linked list of integers
  - Each integer would be a node
  - Integers range from 0 to 99
  - Each node is unique, two duplicate integers won’t exist at the same time
- E.g.
  - 25->3->42->77->10
  - 25->3->42->3->10 won’t happened

Target

- To implement the integer linked list, you are asked to implement 7 functions below
  - InsertBack (int data)
  - DeleteBack()
  - InsertFront(int data)
  - DeleteFront()
  - InsertAfter (int data1, int data2)
  - Delete (int data)
  - Reverse ()

Target

- InsertBack (int data)
  - Insert a data to the end of the linked list
  - E.g. A->B ➔ InsertBack (C) ➔ A->B->C
- DeleteBack()
  - Delete the end node of the linked list
  - E.g. A->B->C ➔ DeleteBack () ➔ A->B

Target

- InsertFront (int data)
  - Insert a data to the front of the linked list
  - E.g. A->B ➔ InsertFront (C) ➔ C->A->B
- DeleteFront()
  - Delete the front node of the linked list
  - E.g. A->B->C ➔ DeleteFront () ➔ B->C

Target

- InsertAfter (int data1, int data2)
  - Insert data2 after data1
  - If data1 doesn’t exist in the linked list, do nothing
  - E.g. A->C ➔ InsertAfter (A, B) ➔ A->B->C
  - E.g. A->C ➔ InsertAfter (D, B) ➔ A->C

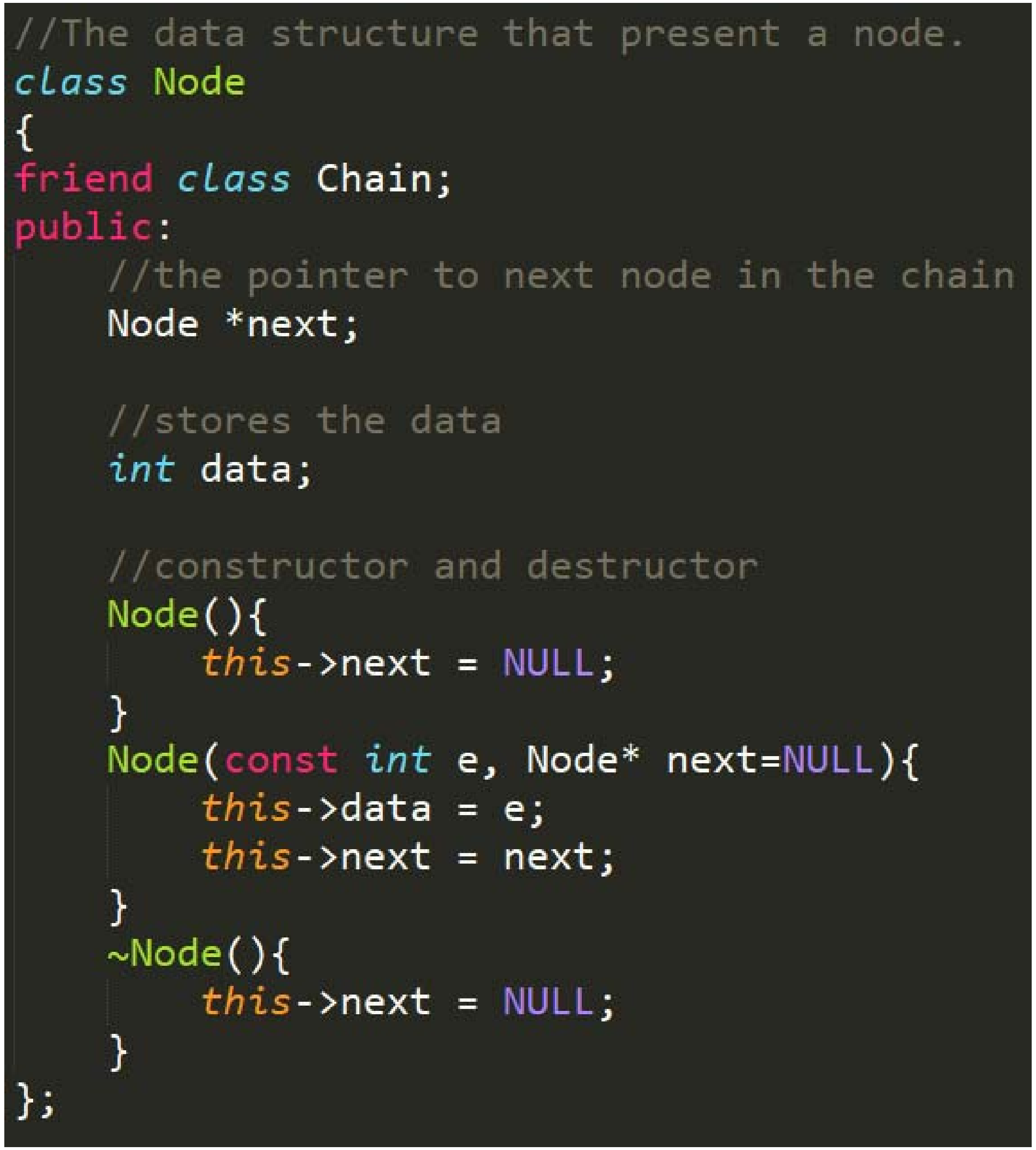
Target

- Delete (int data)
  - Remove the data from the linked list
  - If data doesn’t exist in the linked list, do nothing
  - E.g. A->B->C ➔ Delete (B) ➔ A->C
  - E.g. A->B->C ➔ Delete (Y) ➔ A->B->C
- Reverse ()
  - Reverse the linked list
  - E.g. A->B->C ➔ Reverse () ➔ C->B->A

File Structure

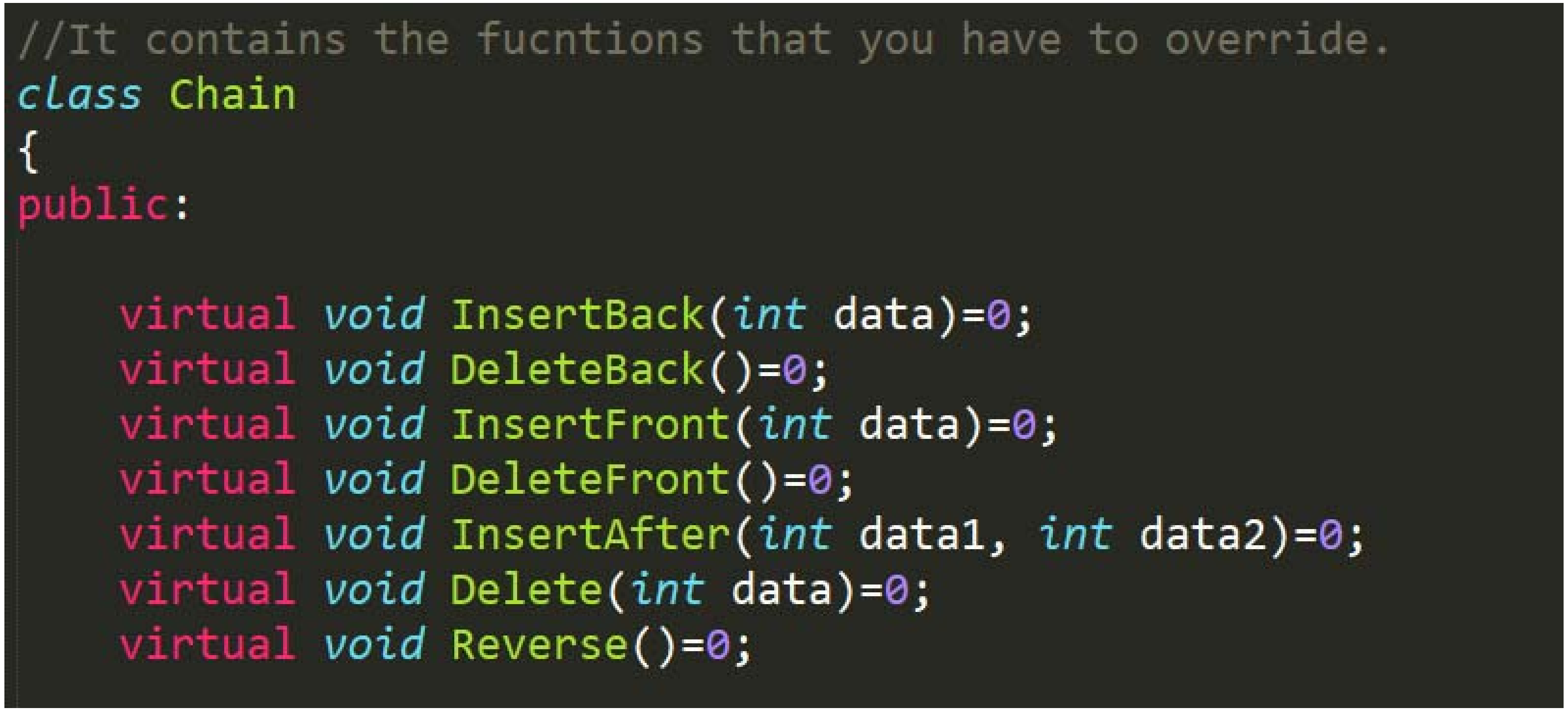
- #include “function.h”
- **class** “Node” represents the DS of a linked list node
- **class** “Chain” represents a linked list
- **class** “Implement” contains your implementation
- Sample input and output

Node



```cpp
//The data structure that present a node.
class Node
{
friend class Chain;
public:
    //the pointer to next node in the chain
    Node *next;

    //stores the data
    int data;

    //constructor and destructor
    Node(){
        this->next = NULL;
    }
    Node(const int e, Node* next=NULL){
        this->data = e;
        this->next = next;
    }
    ~Node(){
        this->next = NULL;
    }
};
```

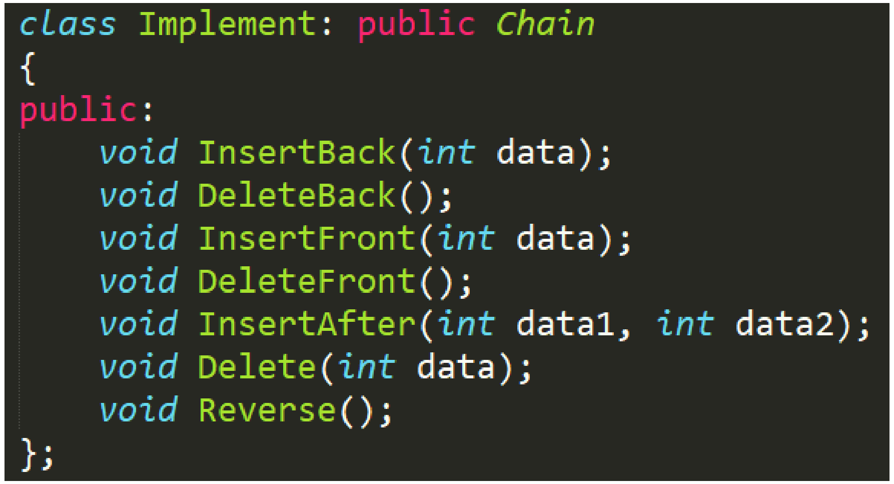
Chain



```
//It contains the fucntions that you have to override.
class Chain
{
public:

    virtual void InsertBack(int data)=0;
    virtual void DeleteBack()=0;
    virtual void InsertFront(int data)=0;
    virtual void DeleteFront()=0;
    virtual void InsertAfter(int data1, int data2)=0;
    virtual void Delete(int data)=0;
    virtual void Reverse()=0;
```

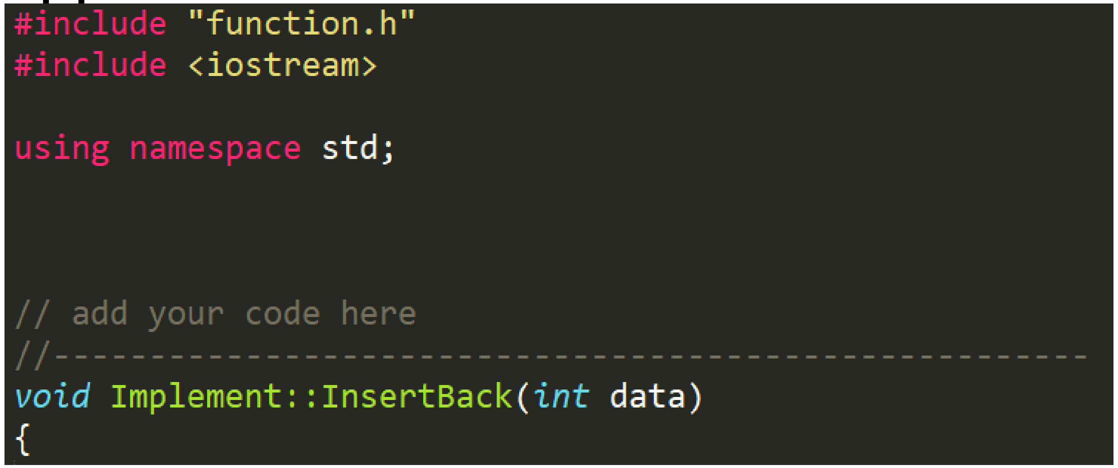
Implement

- In function.h



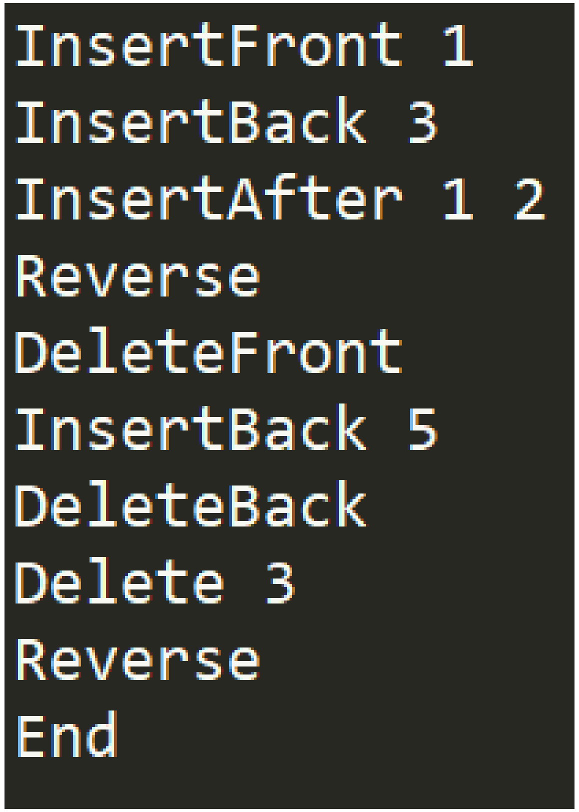
```cpp
class Implement: public Chain
{
public:
    void InsertBack(int data);
    void DeleteBack();
    void InsertFront(int data);
    void DeleteFront();
    void InsertAfter(int data1, int data2);
    void Delete(int data);
    void Reverse();
};
```

- code.cpp

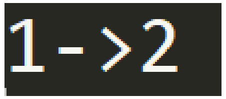


```cpp
#include "function.h"
#include <iostream>

using namespace std;



// add your code here
//------------------------------------------------------
void Implement::InsertBack(int data)
{
```

Sample IO



```
InsertFront 1
InsertBack 3
InsertAfter 1 2
Reverse
DeleteFront
InsertBack 5
DeleteBack
Delete 3
Reverse
End
```



```
1->2
```

STL is not allowed

- <list> <vector> <forward_list> … are not allowed
- If you try to include the above headers, your source files WILL NOT be compiled properly during TA’s evaluation

Submission

- Online Judge: #11627
- Archive your source codes (whole hw2 folder) into a zip file named [studentID]_hw2.zip
  - E.g. 102062999_hw2.zip
- Submit the zip file to ilms system BEFORE the deadline