User Id : weizhe.goh@digipen.edu | Started : 2020.06.25 16:24:18 | Score : 100%

DigiPen

Operator Overloading

Assignment

Rules

Read carefully and check all rules you agree with:

- ☐ Your code must represent your own individual work. Cheating of any kind (copying someone else's work, allowing others to copy your work, collaborating, etc.) will not be tolerated and will be dealt with SEVERELY.
- ☐ Each exercise has description which must be strictly followed.
- ☐ All programs must pass all tests in the main function (when given) to get the final grade. You are not allowed to make any change in the main function in this case.
- ☐ Keep the code proper formatted (correct indentation, max line width is 40 characters).
- ☐ Every week the instructor is available **during the lab time** to discuss following matters:
  - your disagreement with rule in this card,
  - misunderstanding of the current assignment specs,
  - solution for given problems.

Specs

- ☐ Copy the class List from the previous lab. Remove all members that are not used in the following given main function. Fix all problems if any in the rest of the code.
- ☐ Add the "deep" copy constructor to construct a new list by copying all nodes from a list given as the parameter.
- ☐ Same way overload the assignment operator to make the "deep" copy by operator=. (You can use it in the copy constructor for avoiding code duplication.)
- ☐ Overload operator<< to use instead of function print().
- ☐ Your expected output (without ") is "1,2,3
"

★ Code 1 ☑

Run

```
#include <iostream>

struct Node
{
    int data;
    Node* next;
};

class List
{
private:
    Node* head;

public:
    List() :head(nullptr) {}

    List(const List& src) :head(nullptr)
    {
        *this = src;
    }

    List& operator=(const List& src)
    {
        if (this->head != nullptr)
        {
            this->clear();
        }

        Node* current = src.head;
```

Survey

- What is approximate number of hours you spent implementing this assignment?

```
4hrs
```

- Indicate the specific portions of the assignment that gave you the most trouble

```
Understaanding copy constructor,
        while (current)
        {
            this->push_back
            (current->data);
            current = current->next;
        }
        return *this;
    }

    void push_back(int value)
    {
        Node* newNode = new Node;
        Node* current = nullptr;

        newNode->data = value;
        newNode->next = nullptr;

        if (head == nullptr)
        {
            head = newNode;
            return;
        }

        current = head;

        while (current->next)
        {
            current = current->next;
        }
        current->next = newNode;
    }

    void clear()
    {
        Node* deleteNode = nullptr;
        Node* current = nullptr;

        current = head;

        while (current)
        {
            deleteNode = current;
            current = current->next;
            delete deleteNode;
        }

        head = nullptr;
    }

    ~List()
    {
        clear();
    }

    friend std::ostream& operator<<
    (std::ostream& os,const List& list);
};

std::ostream& operator <<
(std::ostream& os, const List& list)
{
    Node* current = nullptr;

    if (list.head == nullptr)
    {
        os << std::endl;
        return os;
    }

    current = list.head;

    while (current)
    {
        if (current->next)
        {
            os << current->data
                << ",";
        }
        else
        {
            os << current->data;
        }
        current = current->next;
    }
    os << std::endl;

    return os;
}
```

```
int main()
{
  List src;
  src.push_back(1);
  src.push_back(2);
  src.push_back(3);

  List dst1(src);

  src.clear();

  List dst2;
  dst2 = dst1;

  std::cout << dst2;
  return 0;
}
```

```
1,2,3
```

By signing this document you fully agree that all information provided therein is complete and true in all respects.	Responder sign: