myTA Hard Copy 7/17/2020, 1:18:29 AM

User Id : weizhe.goh@digipen.edu | Started : 2020.07.16 16:34:29 | Score : 100%

DigiPen

Templates

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

- Implement a new class List based on the one developed in previous assignments with following requirements:

- ☐ The new class List must be a template, so can be used to create lists of different data types.
- ☐ Remove all members from the old class List that are not used in the given main function.
- ☐ Implement members that are missing in your old code. For example, implement new insert() member using existing insertAfter() or insertBefor().
- ☐ Fix all problems if any in the existing old code.
- ☐ Make sure no memory leaks occur.
- ☐ Expected output (without ") is
  "1,2,3
  10.1,20.1
  ".

Code ★ 1 ☑

Run

```
#include <iostream>


template <typename T>
struct Node
{
    T data;
    Node *next;
};

template <typename T>
class List
{
private:
    Node<T> *head;

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

   void push_front(T value)
   {
       Node<T>* newNode = new Node<T>;

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

       head = newNode;
   }
```

Survey

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

```
4 hours
```

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

```
Function template using different
variable
```

```
void push_back(T value)
{
    Node<T>* newNode = new Node<T>;
    Node<T>* 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;
}

List<T>& operator=(const List<T>&rhs)
{
    if (this->head == rhs.head)
    {
        return *this;
    }

    if (this->head)
    {
       clear();
    }

    Node<T>* current = rhs.head;

    while (current)
    {
        this->push_back
        (current->data);
        current = current->next;
    }
    return *this;
}

void insert(int index, T value)
{
    if (index < 0)
        return;

    if (index == 0)
    {
        push_front(value);
        return;
    }

    push_back(value);
    return;
}

friend std::ostream& operator <<
(std::ostream& os,const List<T>&list)
{
    Node<T>* 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;
   }

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

       current = head;

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

       head = nullptr;
   }

   ~List()
   {
       clear();
   }
};
```

```
int main()
{
  List<int> a, b;
  a.push_front(1);
  a.push_back(2);
  a.push_back(3);
  a = a; // test self-assignment
  b = a; // test an assignment
  std::cout << b;

  List<double> c;
  c.insert(0,10.1);
  c.insert(1,20.1);
  std::cout << c;
  return 0;
}
```

```
1,2,3
10.1,20.1
```

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