

2020/03/24 - Doubly-Linked Lists

24 Tháng Ba 2020 3:32 SA

SYNOPSIS

-Welcome back! (well... kinda)

GETTING STARTED

-Copy needed files over:

```
cp ~jplank/cs140/Labs/Lab8/makefile .
```

```
cp -r ~jplank/cs140/Labs/Lab8/include .
```

```
cp -r ~jplank/cs140/Labs/Lab8/src .
```

```
mkdir bin obj
```

-Notice that there is no `dlist.cpp` in `src/`. You write this. I have a template at:

```
~ssmit285/public/code/dlist_template.cpp
```

-Pay attention to the order of the functions specified in the template. This WILL help you greatly.

-Read up the lecture notes for doubly-linked lists. You should be familiar with how they visually work.

SUBMISSION COMMAND

```
tar -cvf lab8.tar src/dlist.cpp
```

SUGGESTED ORDER

- | | |
|------------------------------|-------------------------|
| - Dlist::Dlist() | Default Constructor |
| - Empty, Size | One-Liners |
| - Insert_Before/After | Insertion Handlers |
| - Push_Front/Back | Ditto... |
| - Erase, Clear | Deletion Handlers |
| - Pop_Front/Back | Ditto... |
| - Begin, Rbegin | Iterator Access |
| - End, Rend | Ditto... |
| - Dlist::Dlist(const Dlist&) | "Deep Copy" Constructor |
| - operator=(const Dlist &) | Ditto... |

- I suggest this order to make your life easier.
Let's look at why. Let's look at **Insert_After**:

(Ex. Suppose I have a list L where:

$$L = [s] \leftrightarrow [a] \leftrightarrow [b] \leftrightarrow [c] \leftrightarrow [s]$$

Let's insert "d" AFTER "b"...

Insert_After "d" "b"

1) $[s] \leftrightarrow [a] \leftrightarrow [b] \leftrightarrow [c] \leftrightarrow [s]$

[d]

2) $[s] \leftrightarrow [a] \leftrightarrow [b] \quad [c] \leftrightarrow [s]$

[d] Relink!

$d \rightarrow \text{blink} = b$

$d \rightarrow \text{flink} = b \rightarrow \text{flink} (c)$

$b \rightarrow \text{flink} = d$

$c \rightarrow \text{flink} = d$ (Hint: $d \rightarrow \text{flink} \rightarrow \text{blink}$)

- Now let's look at `Push_Front`...

1) START

`[S] ↔ [a] ↔ [s]`

2) `PUSH_FRONT "b"`

`[S] ↔ [b] ↔ [a] ↔ [s]`

3) `PUSH_FRONT "c"`

`[S] ↔ [c] ↔ [b] ↔ [a] ↔ [s]`

- Notice something? This function simply calls `Insert_After` on `S`, so you get to knock out two functions at once.

- This lab is full of these. So please use the suggested order. It'll help you out.