COP3530 17F Project 1

Kyler Ferrell-Clegg

4991-5518

1087

By submitting this document, I affirm that all the work submitted is solely my own and that in completing this project I did nothing contrary to either the spirit or the letter of the UF Honor Code

SECTION I

	SSLL	PSLL	SDAL	CDAL	CBL
My template class has the required name, file extension, and template parameters. **[y/n]**	Y	Y	Y	Y	Y
My template class was implemented *efficiently* using the required technique (*e.g.*, linked list with a pool of free nodes) as described in Part I and as amended in the announcements. **[y/n]**	Y	Y	Y	Y	Y
I have implemented and throughly tested each of the methods described in Part I and they ***all*** behave correctly. **[y/n]**	Y	Y	Y	Y	N
I have implemented and throughly tested (on one of the official test machines) each of the *big five* member functions and they ***all*** behave correctly. **[y/n]**	N	N	N	N	N
I have implemented and throughly tested each of the *type members* as described in Part II. **[y/n]**	Y	Y	Y	Y	Y
I have implemented and throughly tested *efficient* iterators (both **const** and non-const) over a list instance's data and my list supports the iterator creation operations as described in Part II. **[y/n]**	N	N	N	N	N
I have verified (by testing) that a **const** instance of my list class and the data it holds cannot be modified, either through the list operations nor via iterator over the list's elements. **[y/n]**	N	N	N	N	N
I wrote my tests using CATCH (not required, *except* for the CBL class tests). **[y/n]**	N	N	N	N	Y
I have verified my template class is memory-leak free using valgrind. **[y/n]**	N	N	N	N	N
I certify that all of the responses I have given for this list class are TRUE [*your initials*]	KJF	KJF	KJF	KJF	KJF

SECTION II

The learning experience throughout this project has been very challenging. All the debugging, logic flow, testing, and coding has required a serious amount of effort that has only further improved me as a programmer. I very much enjoyed this project as it was simple in nature yet still required enough effort that it was not just a 1 day walk-in-the-park project. The different type of lists in this project are very intriguing and it was an excellent learning experience to be able to see all the different variations of how data can be stored and memory managed. The SSLL and SDAL were the easiest to complete in my opinion. PSLL was around the medium level. The hardest were then the CDAL and CBL. The CDAL required much more thinking for how the nodes will exist are editing their respective arrays. The CBL was also difficult because of the whole wrap around idea of items in the list which was the biggest challenge to try and hurdle. Overall, I am very satisfied with my final product for this project. This was by far the biggest project in terms of both coding length and required knowledge that I have ever dealt with and I very appreciative of the knowledge I have gained and the overall experience.

SECTION III

I tested SSLL, PSLL, SDAL, & CDAL using main() method while CBL was done using CATCH testing from the provided sample testing.

SSLL

```
Command line:     g++ -o TestSSLL TestSSLL.cpp

FRONT  1
BACK   87
POS 1  53
LENGTH 3
232
45
29
FRONT  232
BACK   29
POS 1  45
LENGTH 3
[232,45,29]
```

PSLL

```
Command line:     g++ -o TestPSLL TestPSLL.cpp

FRONT  1
```

```
BACK   87
POS 1  53
LENGTH 3
232
45
29
FRONT  232
BACK   29
POS 1  45
LENGTH 3
[232,45,29]
```

SDAL

```
Command line:     g++ -o TestSDAL TestSDAL.cpp

FRONT  1
BACK   87
POS 1  53
LENGTH 3
232
45
29
FRONT  232
BACK   29
POS 1  45
LENGTH 3
[232,45,29]
```

CDAL

```
Command line:     g++ -o TestCDAL TestCDAL.cpp

FRONT  1
BACK   87
POS 1  53
LENGTH 3
232
45
29
FRONT  232
BACK   29
POS 1  45
LENGTH 3
[232,45,29]
```

CBL

```
lin309-09:4% make -f Makefile
rm -rf *.o testssll
g++ -c -g -I ./src -std=c++11 test/test_main.cpp
g++ -c -g -I ./src -std=c++11 test/test_SSLL.cpp
g++ -g -o testssll test_main.o test_SSLL.o
./testssll
[A,B,C,D,[A,B,C,D,[A,B,C,D,[A,B,C,D,[A,B,C,D,[A,B,C,D,[A,B,C,D,[A,B,C,D,[A,B,C,D,[A,B,C,D,[A,B,C,
D,[A,B,C,D,==========================================================================
===
All tests passed (16 assertions in 2 test cases)
```