```
/*-------------------------------------------------------------------
Copyright (c) 2014 Author: Jagadeesh Vasudevamurthy
file: slisttest.cpp
you require: ../util/util.cpp ../complex/complex.cpp slisttest.cpp
On linux:
g++  ../util/util.cpp ../complex/complex.cpp slisttest.cpp
valgrind a.out
-- All heap blocks were freed -- no leaks are possible
-------------------------------------------------------------------*/

/*-------------------------------------------------------------------
This file test slist object
DO NOT CHANGE ANY THING IN THIS FILE.
-------------------------------------------------------------------*/

/*-------------------------------------------------------------------
All includes here
-------------------------------------------------------------------*/
#include "slist.h"
#include "../complex/complex.h"

/*-----------------------------------------------------------
static definition - only once at the start
Change to false, if you don't need verbose
-----------------------------------------------------------*/
template <typename T>
bool slist<T>::_display = false;

/*-----------------------------------------------------------
local to this file. Change verbose = true for debugging
-----------------------------------------------------------*/
static bool verbose = false;

/*-------------------------------------------------------------------
printing help
-------------------------------------------------------------------*/
template <typename T>
static void print_help(const char *s, slist<T>& list) {
  cout << s;
  typename slist<T>::iterator itt = list.begin();
  while (itt != list.end()) {
    T& p = *(itt);
    cout << p << " ";
    ++itt;
  }
  cout << endl;
}

/*-------------------------------------------------------------------
store int in linked list
-------------------------------------------------------------------*/
static void test_slist_of_integers(){
  const int N = 10;
  slist<int>::set_display(verbose);
  slist<int> a(NULL, int_descending_order);
  for (int i = 0; i < N; i++) {
    a.append(i);
  }
  print_help("After appending:", a);
  for (int i = 0; i < N; i = i + 2) {
    a.unlink_data(i);
  }
  print_help("After unlinking even items:", a);
  cout << "Printing linked list in reverse order using recusion\n";
  a.print_in_reverse_order_with_recursion();
  cout << "Printing linked list in reverse order without using recusion\n";
```

```
  a.print_in_reverse_order_without_recursion();
  a.reverse();
  print_help("After reverse:", a);
  a.reverse_recur();
  print_help("After reverse recursive:", a);
  bool k = a.find(1);
  if (k) {
    cout << "data 1 is there in the slist. I am going to unlink it \n";
    a.unlink_data(1);
  }
  else {
    cout << "data 1 is NOT there in the slist \n";
  }
  print_help("After unlinking data 1:", a);
  k = a.find(100);
  if (k) {
    a.unlink_data(100);
  }
  print_help("After unlinking data 100:", a);
  k = a.find(9);
  if (k) {
    a.unlink_data(9);
  }
  print_help("After unlinking data 9:", a);
}

/*-------------------------------------------------------------------
store UDT in linked list
-------------------------------------------------------------------*/
static void test_slist_of_udt(){
  const int N = 10;
  slist<complex>::set_display(verbose);
  slist<complex> a(NULL, complex_larger_compare);
  complex c;
  for (int i = 0; i < N; i++) {
    c.setxy(i, i);
    a.append(c);
  }

  print_help("After appending:", a);

  for (int i = 0; i < N; i = i + 2) {
    c.setxy(i, i);
    a.unlink_data(c);
  }
  print_help("After unlinking even items:", a);
  a.reverse();
  print_help("After reverse:", a);
  a.reverse_recur();
  print_help("After reverse recursive:", a);
  c.setxy(1, 1);
  bool k = a.find(c);
  if (k) {
    cout << "data " << c << " is there in the slist. I am going to unlink it \n";
    a.unlink_data(c);
  }
  else {
    cout << "data " << c << " is NOT there in the slist \n";
  }

  print_help("After unlinking data 1+i1:", a);

  c.setxy(100, 100);
  k = a.find(c);
  if (k) {
    a.unlink_data(c);
```

```
133   }
134   print_help("After unlinking data 100+i100:", a);
135   c.setxy(9, 9);
136   k = a.find(c);
137   if (k) {
138     a.unlink_data(c);
139   }
140   print_help("After unlinking data 9+i9:", a);
141 }
142
143 /*-------------------------------------------------------------
144 store pointers to UDT in linked list
145 -------------------------------------------------------------*/
146 static void test_slist_of_ptr_to_udt(){
147   const int N = 10;
148   slist<complex*>::set_display(verbose);
149   slist<complex*> a(delete_complex, complex_larger_compare);
150   for (int i = 0; i < N; i++) {
151     complex* c = new complex(i, i);
152     a.append(c);
153   }
154   print_help("After appending:", a);
155   complex c;
156   for (int i = 0; i < N; i = i + 2) {
157     c.setxy(i, i);
158     a.unlink_data(&c);
159   }
160   print_help("After unlinking even items:", a);
161   a.reverse();
162   print_help("After reverse:", a);
163   a.reverse_recur();
164   print_help("After reverse recursive:", a);
165   c.setxy(1, 1);
166   bool k = a.find(&c);
167   if (k) {
168     cout << "data " << c << " is there in the slist. I am going to unlink it \n";
169     a.unlink_data(&c);
170   }
171   else {
172     cout << "data " << c << " is NOT there in the slist \n";
173   }
174   print_help("After unlinking data 1+i1 ", a);
175
176   c.setxy(100, 100);
177   k = a.find(&c);
178   if (k) {
179     a.unlink_data(&c);
180   }
181   print_help("After unlinking data 100+i100 ", a);
182   c.setxy(9, 9);
183   k = a.find(&c);
184   if (k) {
185     a.unlink_data(&c);
186   }
187   print_help("After unlinking data 9+i9:", a);
188 }
189
190 /*-------------------------------------------------------------
191 Detect loop
192 -------------------------------------------------------------*/
193 static void test_loop(){
194   const int N = 10;
195   slist<int>::set_display(verbose);
196   slist<int> a(NULL, int_descending_order);
197   for (int i = 0; i < N; i++) {
198     a.append(i);
```

```
199   }
200   print_help("After appending:", a);
201   bool p = a.detect_loop();
202   if (p) {
203     cout << "There is a loop in the list \n";
204   }
205   else {
206     cout << "There is NO loop in the list\n";
207   }
208
209   a.create_a_loop(7, 1);
210   //print_help("After making a loop:",a) ; -- LOOP FOR EVER
211   p = a.detect_loop();
212   if (p) {
213     cout << "There is a loop in the list \n";
214   }
215   else {
216     cout << "There is NO loop in the list\n";
217   }
218   //PROGRAM WILL CRASH HERE.
219 }
220
221 /*-------------------------------------------------------------------
222 Properties -> Configuration Properties -> Linker -> System -> Stack Reserve Size 100000000
223
224 Run time Without recursion = 0.005 secs
225 Run time Without recursion = 0.021 secs
226 -------------------------------------------------------------------*/
227 static void test_reverse(){
228   //const int N = 1000000 ; //Stack overflow
229   const int N = 500;
230   slist<int>::set_display(verbose);
231   slist<int> a(NULL, int_descending_order);
232   for (int i = 0; i < N; i++) {
233     a.append(i);
234   }
235   clock_t start1 = clock();
236   a.reverse();
237   clock_t end1 = clock();
238   cout << "Run time Without recursion = " << double(end1 - start1) / CLOCKS_PER_SEC << " secs" << endl;
239
240   clock_t start2 = clock();
241   a.reverse_recur();
242   clock_t end2 = clock();
243   cout << "Run time With recursion = " << double(end2 - start2) / CLOCKS_PER_SEC << " secs" << endl;
244
245 }
246
247 /*-------------------------------------------------------------------
248 main
249 -------------------------------------------------------------------*/
250 int main() {
251   complex::set_display(verbose);
252   test_reverse();
253   test_slist_of_integers();
254   cout << "_____________________________" << endl;
255   test_slist_of_udt();
256   cout << "_____________________________" << endl;
257   test_slist_of_ptr_to_udt();
258   cout << "_____________________________" << endl;
259   if (0) { //Change to 1 to test. The program crash. But OK
260     test_loop();
261   }
262   cout << "_____________________________" << endl;
263   return 0;
264 }
```

```
265
266 //EOF
267
268
```