

```

1  /*
2   * CS 106B, Chris Gregg
3   *
4   * This program contains code to demonstrate pointers.
5   *
6   * NOTE 1: This program will compile with warnings due to uninitialized variables.
7   * Note 2: This program will crash (purposefully) on some choices.
8   */
9
10 #include <iostream>
11 #include "console.h"
12 #include "simpio.h"
13
14 using namespace std;
15
16 // function prototype declarations
17 int requestTest();
18 void addressExample();
19 void crash();
20 void sameVariablePointers();
21 void derefExample();
22 void pointerExample();
23
24 int main() {
25
26     int testNum = requestTest();
27     while (testNum != -1) {
28         switch (testNum) {
29             case 0: // address example
30                 addressExample();
31                 break;
32             case 1:
33                 crash(); // will crash...
34                 break;
35             case 2:
36                 sameVariablePointers();
37                 break;
38             case 3:
39                 derefExample();
40                 break;
41             case 4:
42                 pointerExample();
43             }
44             cout << endl;
45             testNum = requestTest();
46     }
47     cout << "Goodbye!" << endl;
48
49     return 0;
50 }
51
52 int requestTest() {
53     int response = -2; // initial condition
54     cout << "Pointer Examples" << endl;
55     cout << "0. Pointers have addresses" << endl;
56     cout << "1. Seg Fault!" << endl;
57     cout << "2. Pointers to the same variable" << endl;
58     cout << "3. Dereference example" << endl;
59     cout << "4. Pointer example with crash at end." << endl;
60     cout << endl;
61     while (response < -1 or response > 4) {
62         response = getInteger("Please choose an option (0-2), -1 to quit: ");
63     }
64     return response;
65 }
66
67 void addressExample() {
68     string s;
69     int i;
70     double sum;
71
72     cout << "Variables uninitialized" << endl
73         << "    Address of s : " << &s << endl
74         << "    Address of i : " << &i << endl
75         << "    Address of sum: " << &sum << endl

```

```

76     << "      Value of s    : "    << s    << endl
77     << "      Value of i    : "    << i    << endl // will cause warning
78     << "      Value of sum  : "    << sum  << endl // will cause warning
79     << endl;
80
81
82     s    = "Bazinga";
83
84     sum = 0.0;
85     i   = 2;
86     cout << "-----" << endl;
87     cout << "Variables initialized" << endl
88         << "      Address of s : " << &s  << endl
89         << "      Address of i : " << &i  << endl
90         << "      Address of sum: " << &sum << endl
91         << "      Value of s   : " << s   << endl
92         << "      Value of i   : " << i   << endl
93         << "      Value of sum : " << sum << endl
94         << endl;
95
96     while (i < 1000) {
97         sum += i;
98         i *= 2;
99
100        cout << "-----" << endl;
101        cout << "Ready to do loop test again" << endl
102            << "      Address of i : " << &i  << endl
103            << "      Address of sum: " << &sum << endl
104            << "      Value of i   : " << i   << endl
105            << "      Value of sum : " << sum << endl
106            << endl;
107    }
108
109 }
110
111 // This function will attempt to dereference a NULL pointer, which will crash
112 // the program. :(
113 void crash() {
114     string *sPtr = NULL;
115     string s = "hello";
116     cout << *sPtr << endl;
117 }
118
119 void derefExample() {
120     string *sPtr = NULL;
121     string s = "hello";
122     sPtr = &s;
123     *sPtr = "goodbye";
124     cout << *sPtr << endl;
125 }
126
127 void pointerExample() {
128     int      x;
129     int      *p; // declaration of a pointer to an int
130     int      *q; // another pointer to an int
131     int      **z; // declaration of a pointer to a pointer to an int
132
133     /**** STAGE 1 *****/
134
135     cout << endl << "stage 1, initialization" << endl;
136     x = 3;
137     p = &x; //the & operator finds the address of the variable
138
139     cout << "  p: " << p << endl; // print the address of x
140     cout << "  *p: " << *p << endl; // print the value of x;
141     cout << "  x: " << x << endl; // print the value of x;
142
143     /**** STAGE 2 *****/
144     cout << endl;
145     cout << "stage 2, *p = *p - 1 " << endl;
146     *p = *p - 1;
147
148     cout << "  x: " << x << endl; // value of x has decreased by 1
149     cout << "  *p: " << *p << endl; // print the value of x thru p
150

```

```

151     **** STAGE 3 *****/
152     cout << endl;
153     cout << "stage 3, q = p " << endl;
154     q = p;
155
156     cout << " q: " << q << endl;    // should be the same value as p
157     cout << " *q: " << *q << endl;    // which points to x
158
159     **** STAGE 4 *****/
160     cout << endl;
161     cout << "stage 4, *q = *p - 1" << endl;
162     *q = *p - 1;
163
164     // *p and *q and x should all be the same thing
165     // (which is now x = x - 1)
166     cout << " *p: " << *p << endl;
167     cout << " *q: " << *q << endl;
168     cout << " x: " << x << endl;
169
170     **** STAGE 5 *****/
171     cout << endl;
172     cout << "stage 5a, z = &p" << endl;
173     z = &p;
174
175     cout << " z: " << z << endl;
176     cout << " *z: " << *z << endl;
177     cout << "**z: " << **z << endl;
178
179     cout << "stage 5a, z = &q" << endl;
180     z = &q;
181
182     cout << " z: " << z << endl;
183     cout << " *z: " << *z << endl;
184     cout << "**z: " << **z << endl;
185
186     **** STAGE 6 *****/
187     cout << endl;
188     cout << "stage 6, p = (int*) 4" << endl;
189
190     p = (int *) 4;                      // p assigned the location 0x04
191
192     cout << " p: " << p << endl;
193     cout << " q: " << q << endl;
194     cout << "The data stored at location 4 is: " ;
195     cout << *p << endl;                // this will cause a seg fault!
196 }
197
198 void sameVariablePointers() {
199     string *sp1 = NULL;
200     string *sp2 = NULL;
201     string s = "hello";
202     sp1 = &s;
203     cout << *sp1 << endl;
204
205     sp2 = sp1;
206     cout << *sp2 << endl;
207
208     *sp1 = "goodbye";
209     cout << *sp1 << endl;
210     cout << *sp2 << endl;
211 }
```