```
1 //============================================================================
2 // Name        : ICE pointer usage and syntax
3 // Author      :
4 // Version     :
5 // Copyright   : Your copyright notice
6 // Description : Hello World in C++, Ansi-style
7 //============================================================================
8
9 #include <iostream>
10 using namespace std;
11
12 int main() {
13     int* r,rr,rrr,rrrr;      // only r is a pointer
14     int * q,*qq,*qqq,*qqqq; //all pointers
15
16     int a;
17     int* aptr;
18     a = 2;
19     aptr = &a;
20     int b;
21 //  b = aptr;           // error - illegal conversion
22 //  b = &aptr;          // also illegal b isn't a pointer
23     b = *aptr;
24     cout << b << endl;
25     cout << "aptr = " << aptr << endl;
26     aptr=aptr+4;        // legal but questionable?  Adds 16 (10 in hex) - 16 is the
27                         // length of 4 integers
28     cout << "aptr = " << aptr << endl;
29
30     int* myintpointer;                  // pointer variable
31     myintpointer=&a;                    // point it at a
32     cout << myintpointer << endl;       // contents of the pointer - it points to a
33     cout << *myintpointer << endl;      // contents of whatever the pointer variable
34                                         // points to - right now, that integer is
35                                         // uninitialized but that doesn't prevent us
36                                         // from displaying it
37
38     myintpointer = myintpointer + 1;    // increments myintpointer by index 1, actual 4 (int
   length)
39                                         // adds index 1 (4) to myintpointer - now points to the
   field
40                                         // following a in memory.  we told compiler
   myintpointer is a
41                                         // pointer to an int so it acts accordingly
42                                         // how can i add a real 4 to pointer??  Can't.
   Compiler
43                                         // enforces type, and it knows that 4 by itself is just
   a
44                                         // numeric literal, not an address
45
46     cout << myintpointer << endl;       // i can display the new contents of the pointer
47     cout << *myintpointer << endl;      // and also display the contents of whatever it points
   at
48
49     myintpointer++;                     // same as myintpointer=myintpointer+1
50     cout << myintpointer << endl;
51     cout << *myintpointer << endl;
```

```
52
53      myintpointer--;                     // point it back at a...needs two decrements since we
  did
54      myintpointer--;                     // two increments
55      cout << myintpointer << endl;
56
57      *myintpointer++;                    // danger - this also adds index 1 (4) to myintpointer.
58                                          // has to do with the bind characteristics of the
59                                          // ++ and -- operators.  one would think that this
  would
60                                          // increment the contents of whatever the pointer
  points to,
61                                          // but it doesn't
62      cout << myintpointer << endl;
63      cout << *myintpointer << endl;
64
65      *myintpointer--;                    // point it back at a
66      *myintpointer = *myintpointer + 1;  // a started at 2 from line 18, now it should now be 3
67      cout << myintpointer << endl;
68      cout << *myintpointer << endl;
69
70      (*myintpointer)++;                  // adds 1 to a; should now be 4. parens override
  default ++ bind.
71      cout << *myintpointer << endl;      // should be 4 now
72
73      return 0;
74  }
75
```

Sample output:

```
2
0x62fe08
6487560
0x62fe0c
0
0x62fe04
0x62fe08
6487560
0x62fe04
3
4
```