```
EffectiveCPP.cpp
```

```
2
  .
.
3
5
  #include <iostream>
6
  using namespace std;
7
8
9
10
  class CPoint
  11
12
  private:
13
    static int iCtorCount;
14
15
    static int iDtorCount;
16
    static int iCopyCount;
17
    static int iAssignCount;
18
19
  public:
    int iX;
20
    int iY;
2.1
2.2
23
  public:
24
    CPoint() : iX(0), iY(0) {iCtorCount ++;}
    CPoint(int x, int y) : iX(x), iY(y) {iCtorCount++;}
25
    CPoint(const CPoint& Source);
26
27
    CPoint& operator=(const CPoint& Source);
28
    ~CPoint() {iDtorCount++;}
29
    static void listCount(void)
30
31
32
       cout << "--> Call Count: "
       << iCtorCount + iDtorCount + iCopyCount + iAssignCount
33
34
          (Ctor: " << iCtorCount
       << ", Dtor: " << iDtorCount
35
      << ", Copy: " << iCopyCount
36
       << ", Assign: " << iAssignCount << ") " << endl;
37
38
39
  40
41
42
43
44
  45
46
  int CPoint::iCtorCount = 0;
47
  int CPoint::iCopyCount = 0;
  int CPoint::iDtorCount = 0;
48
49
  int CPoint::iAssignCount = 0;
50
51
52
53
  CPoint::CPoint(const CPoint& Source)
54
55
56
     *this = Source;
57
     iCopyCount++;
58
  59
60
61
62
  63
64
  CPoint& CPoint::operator=(const CPoint& Source)
65
66
     iX = Source.iX;
67
     iY = Source.iY;
68
     iAssignCount++;
     return *this;
69
70
  71
72
```

```
73
74
75
   class CLine
   76
77
78
   private:
79
     CPoint P1;
80
     CPoint P2;
81
   public:
82
     CLine (const CPoint p1, const CPoint p2);
83
84
     void List(void);
85
     86
87
88
89
90
   CLine::CLine(const CPoint p1, const CPoint p2)
   91
92
93
     P1 = p1;
94
     P2 = p2;
95
96
   97
98
99
100
   void CLine::List(void)
   101
102
103
     cout << "Line Object:" << endl;</pre>
                             iY = " << P1.iY << endl;
iY = " << P2.iY << endl << endl;</pre>
     cout << " P1: iX = " << P1.iX << "
104
105
     cout << " P2: iX = " << P2.iX << "
106
   107
108
109
110
111
   int main(void)
   112
113
114
      // show initial call count
115
      CPoint::listCount();
      cout << endl;</pre>
116
117
118
      // definitions
      CLine* pL1;
CPoint P1;
119
120
      CPoint P2 (30, 30);
121
122
      // dynamic instantiation of a line object
123
124
      pL1 = new CLine(P1, P2);
125
      // list values of line
126
127
      pL1->List();
128
      // show final call count
129
      CPoint::listCount();
130
131
      cout << endl;
132
      // delete dynamic line object
133
134
      delete pL1;
135
136
      return 0;
137
138
```

EffectiveCPP.cpp