

```

1 // Demonstrate constructors and destructors
2 // George F. Riley, Georgia Tech, Spring 2009
3
4 #include <iostream>
5
6 using namespace std;
7
8 class A
9 {
10 public:
11     // Define three constructors
12     A();
13     A(int a0);
14     A(const A& a0);
15     // Define the destructor
16     ~A();
17
18 public:
19     int a; // Member variable
20 };
21
22 // Default Constructor
23 A::A()
24     : a(0)
25 {
26     cout << "Default constructor, a = " << a << endl;
27 }
28
29 // Non-Default Constructor
30 A::A(int a0)
31     : a(a0)
32 {
33     cout << "Int constructor, a = " << a << endl;
34 }
35
36 // Copy Constructor
37 A::A(const A& a0)
38     : a(a0.a)
39 {
40     cout << "Copy constructor, a = " << a << endl;
41 }
42
43 // Destructor
44 A::~A()
45 {
46     cout << "Destructor, a = " << a << endl;
47 }
48
49
50
51 // Define some global variables of class A
52 A ga1;
53 A ga2(2);
54 A ga3(ga2); // This compiles, but is not a good idea. Why?
55
56 // Subroutine with no arguments

```

Program ConstructorsDestructors.cc

```

57 void sub1()
58 {
59     cout << "Entering sub1" << endl;
60     A sub1a;
61     A sub1a2(2);
62     A sub1a3(sub1a2);
63     cout << "Exiting sub1" << endl;
64 }
65
66 // Subroutine with object A passed by value
67 void sub2(A a)
68 {
69     cout << "Entering sub2" << endl;
70     A sub2a(a);
71     cout << "Exiting sub2" << endl;
72 }
73
74 // Subroutine with object A passed by address
75 void sub3(A* a)
76 {
77     cout << "Entering sub3" << endl;
78     A sub2a(3);
79     cout << "Exiting sub3" << endl;
80 }
81
82 int main()
83 {
84     cout << "Entering main" << endl;
85     A a0;
86     A a1(a0);
87     A a2(2);
88     A a3(3);
89     A a4 = a2; // What about this one?
90     cout << "Calling sub1" << endl;
91     sub1();
92     cout << "Back from sub1" << endl;
93     cout << "Calling sub2" << endl;
94     sub2(a3);
95     cout << "Back from sub2" << endl;
96     cout << "Calling sub3" << endl;
97     sub3(&a2);
98     cout << "Back from sub3" << endl;
99     cout << "Exiting main" << endl;
100 }
101
102
103
104
105
106
107

```

Program ConstructorsDestructors.cc (continued)

```
1 Default constructor, a = 0
2 Int constructor, a = 2
3 Copy constructor, a = 2
4 Entering main
5 Default constructor, a = 0
6 Copy constructor, a = 0
7 Int constructor, a = 2
8 Int constructor, a = 3
9 Copy constructor, a = 2
10 Calling sub1
11 Entering sub1
12 Default constructor, a = 0
13 Int constructor, a = 2
14 Copy constructor, a = 2
15 Exiting sub1
16 Destructor, a = 2
17 Destructor, a = 2
18 Destructor, a = 0
19 Back from sub1
20 Calling sub2
21 Copy constructor, a = 3
22 Entering sub2
23 Copy constructor, a = 3
24 Exiting sub2
25 Destructor, a = 3
26 Destructor, a = 3
27 Back from sub2
28 Calling sub3
29 Entering sub3
30 Int constructor, a = 3
31 Exiting sub3
32 Destructor, a = 3
33 Back from sub3
34 Exiting main
35 Destructor, a = 2
36 Destructor, a = 3
37 Destructor, a = 2
38 Destructor, a = 0
39 Destructor, a = 0
40 Destructor, a = 2
41 Destructor, a = 2
42 Destructor, a = 0
```

Program ConstructorsDestructors-output.txt