MA122 - Computer Programming and Applications

Indian Institute of Space Science and Technology

April 05, 2017

Lecture 26

example 1


```
1  #include <iostream>
2  using namespace std;
3
4  class Distance
5  {
6      private:
7          int meter;
8      public:
9          Distance(){meter=0;}
10     //Distance(): meter(0) { }
11         //friend function
12         friend int addFive(Distance);
13 };
```

example 1


```
1  // friend function definition
2  int addFive(Distance d)
3  {
4      //accessing private data from non-member function
5      d.meter += 5;
6      return d.meter;
7  }
8
9  int main()
10 {
11     Distance D;
12     cout<<"Distance: "<< addFive(D);
13     return 0;
14 }
```

example 2


```
1  #include <iostream>
2  using namespace std;
3
4  // forward declaration
5  class B;
6  class A {
7  private:
8      int numA;
9  public:
10     A(): numA(12) { }
11     // friend function declaration
12     friend int add(A, B);
13 };
```

example 2


```
1
2 class B {
3 private:
4     int numB;
5 public:
6     B(): numB(1) { }
7     // friend function declaration
8     friend int add(A , B);
9 };
```

example 2


```
1  // Function add() is the friend function of classes A
       and B
2  // that accesses the member variables numA and numB
3  int add(A objectA, B objectB)
4  {
5      return (objectA.numA + objectB.numB);
6  }
7
8  int main()
9  {
10     A objectA;
11     B objectB;
12     cout<<"Sum: "<< add(objectA, objectB);
13     return 0;
14 }
```

example 3


```
1  #include <iostream>
2  class B;
3  class A
4  {
5  public:
6      void showB(B& );
7  };
8
9  class B
10 {
11 private:
12     int b;
13 public:
14     B() { b = 2; }
15     friend void A::showB(B& x); // Friend function
16 };
```

example 3


```
1  void A::showB(B &x)
2  {
3      // Since show() is friend of B, it can
4      // access private members of B
5      std::cout << "B::b = " << x.b<<std::endl;
6  }
7
8  int main()
9  {
10     A a;
11     B x;
12     a.showB(x);
13     return 0;
14 }
```

Lecture 26

example 1


```
1  #include <iostream>
2  using namespace std;
3  class Box {
4  public:
5      static int objectCount;
6      // Constructor definition
7      Box(double l = 2.0, double b = 2.0, double h =
           2.0) {
8          cout <<"Constructor called." << endl;
9          length = l;
10         breadth = b;
11         height = h;
12         // Increase every time object is created
13         objectCount++;
14     }
15     double Volume() {
16     return length * breadth * height;
17     }
```

example 1


```
1  private:
2      double length;    // Length of a box
3      double breadth;   // Breadth of a box
4      double height;    // Height of a box
5  };
6
7  // Initialize static member of class Box
8  int Box::objectCount = 0;
9
10 int main(void) {
11     Box Box1(3.3, 1.2, 1.5); // Declare box1
12     Box Box2(8.5, 6.0, 2.0); // Declare box2
13
14     // Print total number of objects.
15     cout << "Total objects: " << Box::objectCount <<
           endl;
16     return 0;
17 }
```

example 2


```
1  #include <iostream>
2
3  using namespace std;
4
5  class Box {
6  public:
7      static int objectCount;
8      // Constructor definition
9      Box(double l = 2.0, double b = 2.0, double h =
           2.0) {
10         cout <<"Constructor called." << endl;
11         length = l;
12         breadth = b;
13         height = h;
14         // Increase every time object is created
15         objectCount++;
16     }
```

example 2


```
1
2       double Volume() {
3           return length * breadth * height;
4       }
5
6       static int getCount() {
7           return objectCount;
8       }
9
10  private:
11      double length;    // Length of a box
12      double breadth;   // Breadth of a box
13      double height;    // Height of a box
14  };
```

example 2


```
1  // Initialize static member of class Box
2  int Box::objectCount = 0;
3  int main(void) {
4
5      // Print total number of objects before creating
           object.
6      cout << "Inital Stage Count: " << Box::getCount()
           << endl;
7
8      Box Box1(3.3, 1.2, 1.5); // Declare box1
9      Box Box2(8.5, 6.0, 2.0); // Declare box2
10
11     // Print total number of objects after creating
           object.
12     cout << "Final Stage Count: " << Box::getCount()
           << endl;
13     return 0;
14 }
```

example 3


```
1  #include <iostream>
2  using namespace std;
3  class A
4  {
5  public:
6      A() { cout << "A's Constructor Called " << endl; }
7  };
8  class B
9  {
10     static A a;
11 public:
12     B() { cout << "B's Constructor Called " << endl; }
13 };
14 int main()
15 {
16     B b;
17     return 0;
18 }
```

example 4


```
1  #include <iostream>
2  using namespace std;
3  class A
4  {
5      int x;
6  public:
7      A() { cout << "A's constructor called " << endl; }
8  };
9  class B
10 {static A a;
11 public:
12     B() { cout << "B's constructor called " << endl; }
13     static A getA() { return a; }
14 };
15 int main(){
16     B b;
17     A a = b.getA();
18     return 0;}
```

example 5


```
1  #include <iostream>
2  using namespace std;
3  class A
4  {int x;
5  public:
6      A() { cout << "A's constructor called " << endl; }
7  };
8  class B
9  {static A a;
10 public:
11     B() { cout << "B's constructor called " << endl; }
12     static A getA() { return a; }
13 };
14 A B::a; // definition of a
15 int main()
16 {   B b1, b2, b3;
17     A a = b1.getA();
18     return 0;}
```