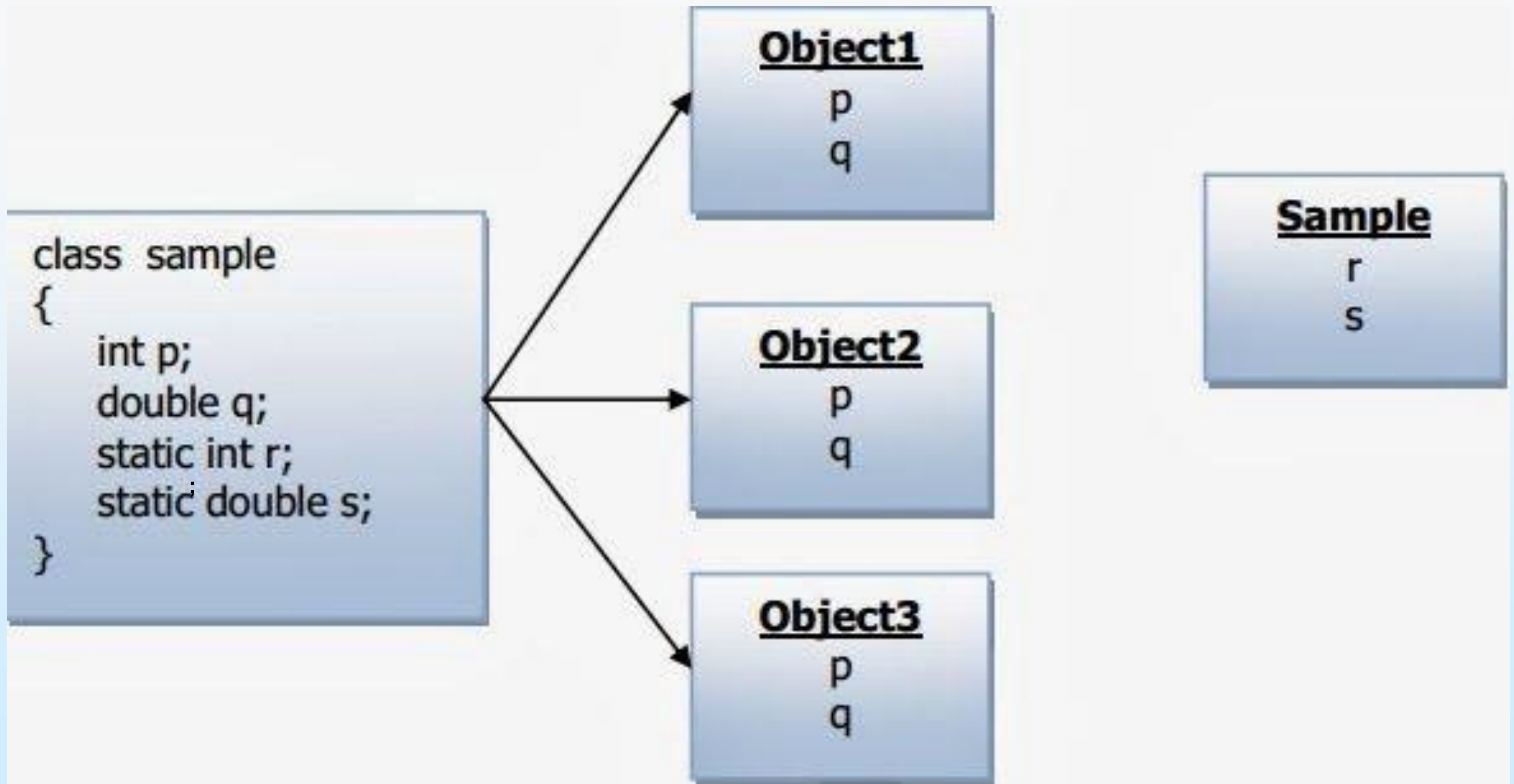


Access Control

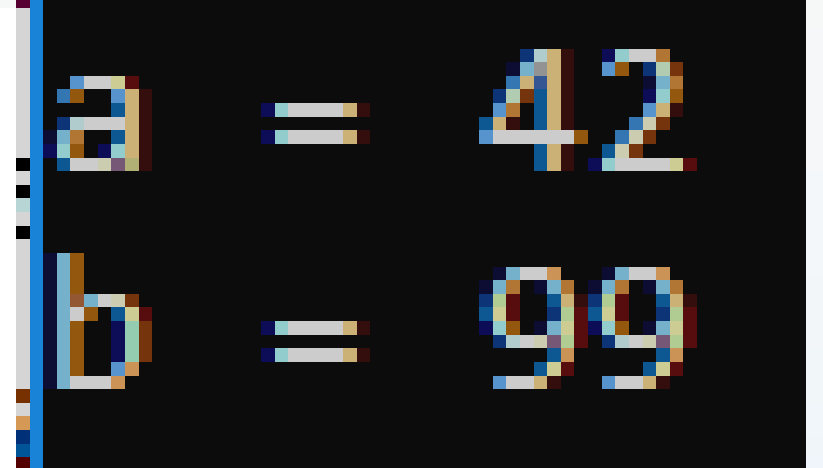
□ Static members of a class

- Can be accessed without reference to any object
 - ▶ E.g., main() method (called before any objects exist)
- Instance variables can be static
 - ▶ Act like global variables
 - ▶ When objects are created, no copy of static variables is made
 - ▶ All objects share the same static variable
- Methods can be static
 - ▶ They can only call other static methods
 - ▶ They can access only static data
 - ▶ They cannot refer to *this* or *super* in any way
- If computation is required to initialize static variable
 - ▶ Declare a static block (gets executed only once, when the class is first loaded)



Example: static

```
1  class StaticDemo
2  {
3      static int a = 42;
4      static int b = 99;
5      static void callme()
6      {
7          System.out.println("a = " + a);
8      }
9  }
10
11 class StaticByName
12 {
13     public static void main(String args[])
14     {
15         StaticDemo.callme();
16         System.out.println("b = " + StaticDemo.b);
17     }
18 }
```



a = 42
b = 99

```
1  class UseStatic
2  {
3      static int a = 3;
4      static int b;
5
6      static void meth(int x)
7      {
8          System.out.println("x = " + x);
9          System.out.println("a = " + a);
10         System.out.println("b = " + b);
11     }
12
13     static
14     {
15         System.out.println("Static block initialized.");
16         b = a * 4;
17     }
18
19     public static void main(String args[])
20     {
21         meth(42);
22     }
23 }
```

```
Static block initialized.
x = 42
a = 3
b = 12
```

Question-1

```
1  class Static_Test
2  {
3      int denom = 3;
4      static int val = 1024;
5
6      static int valDivDenom()
7      {
8          return val/denom;
9      }
10 }
11 class Static_Demo
12 {
13     public static void main(String[] args)
14     {
15         System.out.println("val is " +Static_Test.valDivDenom() );
16     }
17 }
```

Question-1 (Ans)

```
1  class Static_Test
2  {
3      int denom = 3;
4      static int val = 1024;
5
6      static int valDivDenom()
7      {
8          return val/denom; // Error !
9      }
10 }
11 class Static_Demo
12 {
13     public static void main(String[] args)
14     {
15         System.out.println("val is " +Static_Test.valDivDenom() );
16     }
17 }
```

Question-2

```
1  class StaticBlock
2  {
3      static double a;
4      static double b;
5
6      static
7      {
8          System.out.println("Inside static block.");
9          a = Math.sqrt(25.0);
10         b = Math.sqrt(49.0);
11     }
12
13     StaticBlock(String msg)
14     {
15         System.out.println(msg);
16     }
17 }
18
```

```
Inside static block.
Inside Constructor
5.0
7.0
```

```
19 class SDemo
20 {
21     public static void main(String[] args)
22     {
23         StaticBlock ob = new StaticBlock("Inside Constructor");
24
25         System.out.println( StaticBlock.a );
26         System.out.println( StaticBlock.b );
27     }
28 }
```

Question-3:

```
1  class MyClass
2  {
3      int count = 0;
4
5      MyClass()
6      {
7          count++;
8      }
9  }
10
11 class Test
12 {
13     public static void main(String[] args)
14     {
15         for(int i=0; i < 3; i++)
16         {
17             MyClass obj = new MyClass();
18             System.out.println("Number of objects created: " + obj.count);
19         }
20     }
21 }
```

```
Number of objects created: 1
Number of objects created: 1
Number of objects created: 1
```