

### Ví dụ về biến tĩnh static: *Counter1.java*

```
1 public class Counter1 {
2     static int count = 0;
3     public void pt1()
4     {
5         count++;
6         System.out.println(count);
7     }
8     public static void main(String args[]) {
9         Counter1 c1 = new Counter1(); c1.pt1();
10        Counter1 c2 = new Counter1(); c2.pt1();
11        Counter1 c3 = new Counter1(); c3.pt1();
12    }
13 }
14 // 1
15 // 2
16 // 3
```

### Ví dụ về biến static: *Student.java*

```
3 public class Student {
4     int id;
5     String name;
6     static String college = "Đại học Bách Khoa";
7     // phương thức khởi tạo Constructor
8     public Student(int id, String name) {
9         this.id = id;
10        this.name = name;
11    }
12    void display() {
13        System.out.println(id + " - " + name + " - " + college);
14    }
15    public static void main(String args[]) {
16        Student s1 = new Student(888, "MaiHa");
17        Student s2 = new Student(999, "HaoMinh");
18        s1.display(); s2.display();
19    }
}
```

### Ví dụ về phương thức static truy cập biến tĩnh static: *Student1.java*

```
1 public class Student1 {
2     int id;
3     String name;
4     static String college = "Đại học Bách Khoa"; // biến static
5     static void change() {
6         college = "ĐHBK";
7     }
8     Student1(int id, String name) {
9         this.id = id; this.name = name;
10    }
11    void display() {
12        System.out.println(id + " - " + name + " - " + college);
13    }
14    public static void main(String args[]) {
15        Student1.change(); // thay đổi giá trị biến static
16        Student1 s1 = new Student1(888, "TueNhi");
17        Student1 s2 = new Student1(999, "TueNghi");
18        s1.display(); s2.display();
19    }
}
```

### Ví dụ về phương thức static truy cập biến non-static: *Disadvantage.java*

```
1 class Disadvantage {
2     int a = 40; // non-static
3     public static void show()
4     {
5         System.out.println(a); // lỗi
6     }
7     public static void main(String args[]) {
8         Disadvantage dt = new Disadvantage();
9         dt.show();
10    }
11 }
12 // Compile Time Error
```

Ví dụ về khối lệnh static: *Student2.java*

```
1 public class Student2 {
2     static {
3         System.out.println("Khoi static: Hello !");//khoi lệnh {...}
4     }
5
6     public static void main(String a[])
7     {
8         System.out.println("Main: Xin chao !");
9     }
10 }
11 // Khoi static: Hello !
12 // Main: Xin chao !
```

Ví dụ về khối lệnh static không có phương thức main(): *Student3.java*

```
1 public class Student3 {
2     static {
3         System.out.println("static block is invoked");
4         System.exit(0);
5     }
6 }
7
8 // TH < jdk7: static block is invoked
9 // TH >= jdk7: Error
10 // Main method not found in class Student3, please define the main
11 // public static void main(String[] args)
```