

```
1 package constructor15_01;
2 //import constructor_15.ObjectClass;
3
4 public class MainClass {
5     public static void main(String[] args) {
6         ObjectClass object1 = new ObjectClass();
7     }
8 }
9 }
10
```

```
1 package constructor15_01;  
2  
3 public class ObjectClass {  
4  
5 }  
6
```

```
1 package constructor15_02;
2
3 public class MainClass {
4     public static void main(String[] args) {
5         ObjectClass object1 = new ObjectClass(10);
6
7         int[] iArr = {10, 20, 30};
8         ObjectClass object2 = new ObjectClass("Hello", iArr);
9     }
10 }
11 // 사용자 정의 생성자가 있으면, 컴파일러는 default 생성자를 만들지 않는다.
12 //UserDefined constructor with int x
13 //     -- ObjectClass() --
14 //     s ----> Hello
15 //     iArr ----> [I@1b6d3586    <-- 배열은 객체이므로 주소가 찍혔다.
```

```
1 package constructor15_02;
2
3 public class ObjectClass {
4
5     public ObjectClass(int x) {
6         System.out.println("UserDefined constructor with int x");
7     }
8
9     // 매개변수는 필요한 만큼 얼마든지 만들어서 사용할 수 있다.
10    public ObjectClass(String s, int[] iArr) {
11        System.out.println("-- ObjectClass() --");
12        System.out.println("s ----> " + s);
13        System.out.println("iArr ----> " + iArr);
14    }
15 }
16
```

```

1 package constructor15_03_exmaple;
2
3 public class MainClass {
4
5     public static void main(String[] args) {
6
7         // 디폴트 생성자
8         ObjectEx obj1 = new ObjectEx();
9
10        System.out.println(); // 한 줄 개행
11
12        // 사용자 정의 생성자
13        ObjectEx obj2 = new ObjectEx(10);
14
15        int arr[] = {10, 20, 30};
16        ObjectEx obj3 = new ObjectEx("Java", arr);
17
18
19        // 소멸자
20        ObjectEx obj4;
21
22        obj4 = new ObjectEx();
23        System.out.println(obj4);
24        obj4 = new ObjectEx(); // 위에 먼저 생성된 객체와의 관계는 끊고,
25                               // 두 번째 객체와 연결이 된다.
26                               // 관계가 끊어진 첫 번째 객체는 언젠가 GC가 나타나서 회수한다.
27                               // 이때 호출되는 게 finalize() 메서드이다.
28        System.out.println(obj4);
29
30        // Default constructor
31        // constructor15_03_exmaple.ObjectEx@4554617c <- 첫 번째 생성된 객체
32        // Default constructor
33        // constructor15_03_exmaple.ObjectEx@74a14482 <- 두 번째 생성된 객체
34
35        System.gc(); // 거의 사용되지 않는다. 사용한다고 해서 바로 작동하지 않는다.
36
37        // this 키워드
38        int ints[] = {10, 20, 30};
39        ObjectEx obj5 = new ObjectEx(10, "Java", ints);
40    }
41
42 }
43
44 //Default constructor
45 //
46 //Custom constructor
47 //num = 10
48 //UserDefined constructor
49 //String: Java
50 //nums: [I@1b6d3586
51 //Default constructor <-- 첫 번째 만들어진 객체.
52 //constructor15_03_exmaple.ObjectEx@4554617c
53 //Default constructor
54 //constructor15_03_exmaple.ObjectEx@74a14482
55 //UserDefined constructor
56 //-- finalize() method -- <-- 뭐가 제거될 때 호출된 것인가?
57 // <-- 첫 번째 생성된 객체가 소멸될 때 호출된 소멸자이다.

```

```

1 package constructor15_03_exmaple;
2
3 public class ObjectEx {
4
5     int num;
6     String str;
7     int nums[];
8
9     public ObjectEx() {
10
11         System.out.println("Default constructor");
12
13     }
14
15     public ObjectEx(int i) {
16
17         System.out.println("Custom constructor");
18         num = i;
19         System.out.println("num = " + num);
20
21     }
22
23     public ObjectEx(String s, int i[]) {
24
25         System.out.println("UserDefined constructor");
26         str = s;
27         nums = i; // 배열(?)
28
29         System.out.println("String: " + str);
30         System.out.println("nums: " + nums);
31
32     }
33
34     public ObjectEx(int i, String s, int is[]) {
35
36         System.out.println("UserDefined constructor");
37
38         this.num = i;
39         this.str = s;
40         this.nums = is;
41
42     }
43
44     // finalize()는 모든 클래스에 명시하지 않아도 된다.
45     // 명시하는 경우가 거의 없다. 이런 게 있다는 것만 알고 넘어가면 된다.
46     @Override
47     protected void finalize() throws Throwable {
48
49         System.out.println(" -- finalize() method --");
50
51         super.finalize();
52     }
53
54 }
55

```

```
1 package constructor15_04_this;
2
3 public class MainClass {
4     public static void main(String[] args) {
5
6         ObjectClass object1 = new ObjectClass(10, 20);
7
8         object1.getInfo();
9
10    }
11 }
12
```

```
1 package constructor15_04_this;
2
3 public class ObjectClass {
4
5     public int xNum;
6     public int yNum;
7
8     public ObjectClass(String s, int[] iArr) {
9         System.out.println("-- ObjectClass() --");
10
11         System.out.println("s ----> " + s);
12         System.out.println("iArr ----> " + iArr);
13     }
14
15     public ObjectClass(int xNum, int yNum) {
16
17         this.xNum = xNum; // 전역변수인지 매개변수인지 헤깔리지 않기 위해 this를
        넣어줬다.
18         this.yNum = yNum;
19
20     }
21
22     public void getInfo() {
23         System.out.println("xNum ----> " + xNum); // 여기에서는 굳이 this.xNum을 쓸
        필요는 없다.
24         System.out.println("yNum ----> " + yNum); // 헤깔리지 않기 때문이다.
25     }
26
27 }
28
```