



TECHPROED

PROFESSIONAL TECHNOLOGY EDUCATION

WELCOME TO TECHPROED JAVA TUTORIAL

Testi baslatmak için asagidaki adımları takip ediniz

Go to www.socrative.com

Click on Login

Click on Student Login

Room Name: ALPTEKIN3523

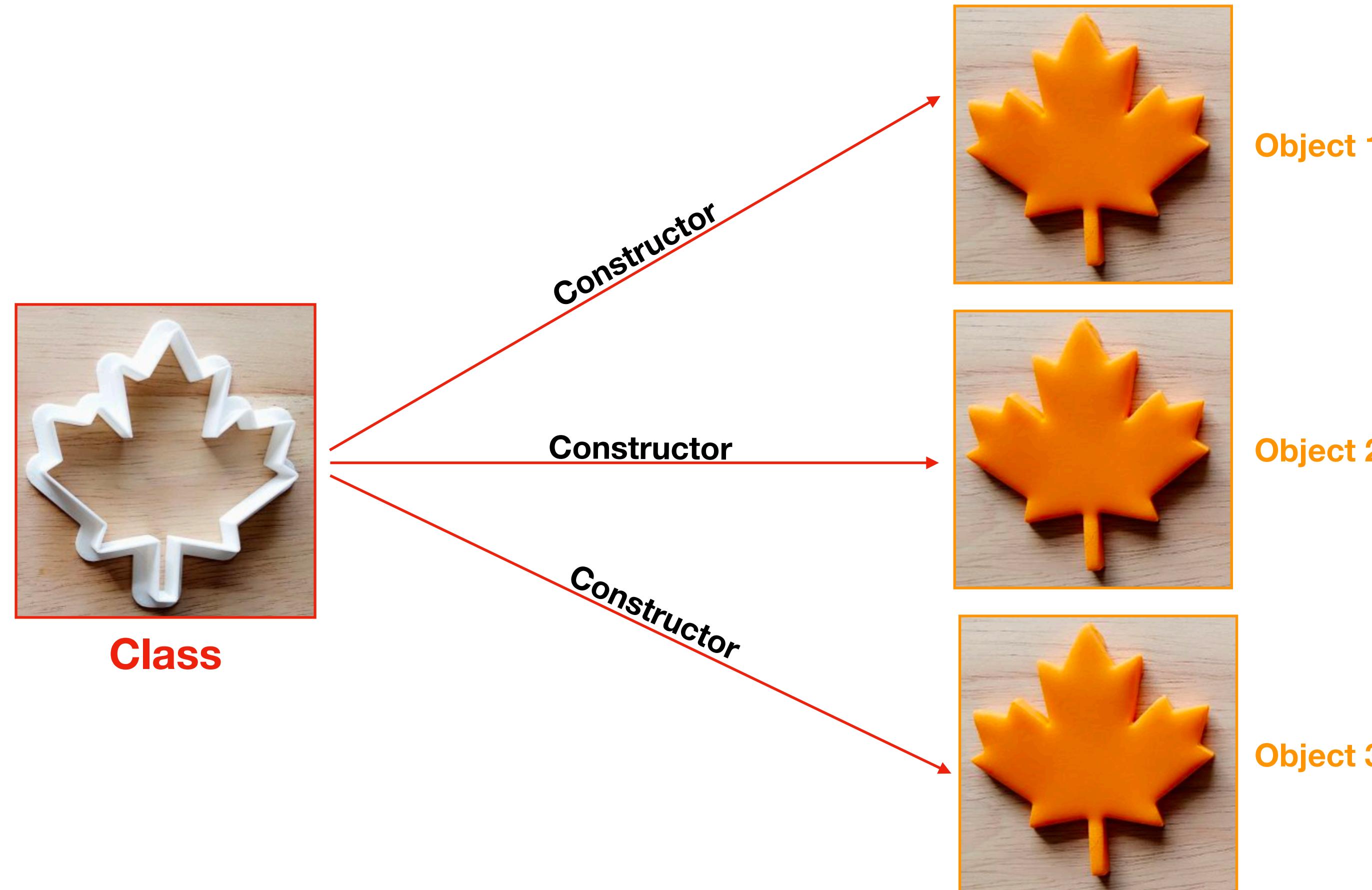
Kayıtta kullandığınız ismi tam olarak yazınız

Time: 11 Minutes



Object nasıl üretilir ?

Constructor method degildir, variable degildir. Constructor bir Java kodudur ve Class'dan obje üretmek için kullanılır..



Constructor nedir?

```
public class MyClass {  
    MyClass() {  
    }  
}
```

- 1) Constructor class'ın içinde olmalıdır
- 2) Constructor 'in ismi class'ın ismi ile aynı olmalıdır.
- 3) Constructor'larda return type olmaz



Note 1: Constructor is not a method, because it has no return type.

Therefore, do not mention “constructor method”, mention just “constructor” while you talk about constructor

Note 2: When we create a class, Java creates a “default constructor” automatically for the class. So we can create objects by using every class without creating any constructor.

Note 3: Default constructors have no any parameters

```
MyClass() {  
}  
}
```

Note 4: When we create a constructor by ourselves, default constructor is cancelled by Java

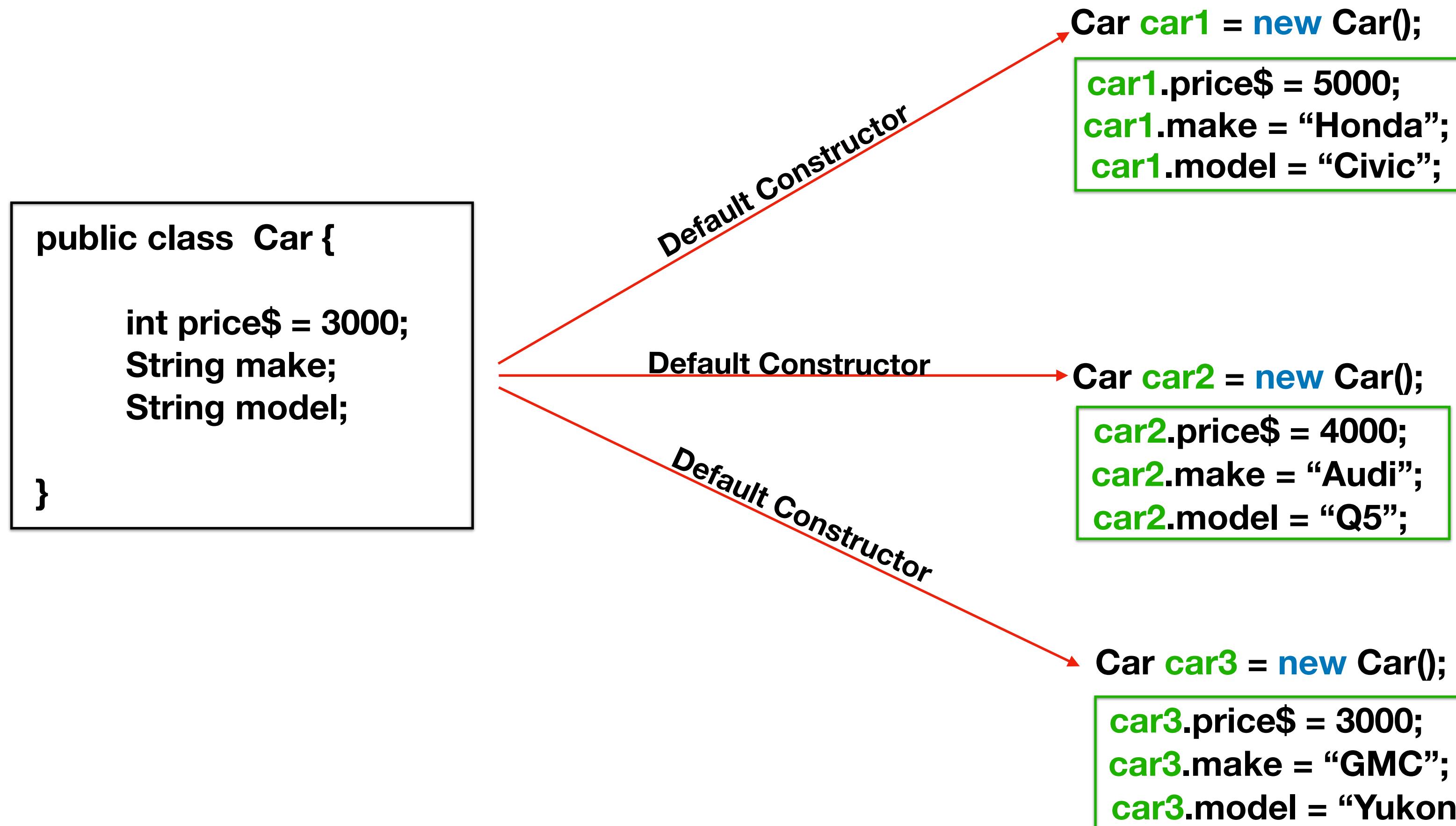
```
MyClass() {  
}  
}
```

```
MyClass(String str) {  
}  
}
```

```
MyClass(int i, String str) {  
}  
}
```



How does a constructor work ?



```
public class Car {
```

```
    int price$;  
    String make;  
    String model;
```

```
    public Car(){  
        this.price$;  
        this.make = "Honda";  
        this.model;  
    }
```

```
}
```

No Parameters Constructor

No Parameters Constructor

No Parameters Constructor

```
Car car1 = new Car();
```

```
car1.price$ = 5000;  
car1.make = "Honda";  
car1.model = "Civic";
```

```
Car car2 = new Car();
```

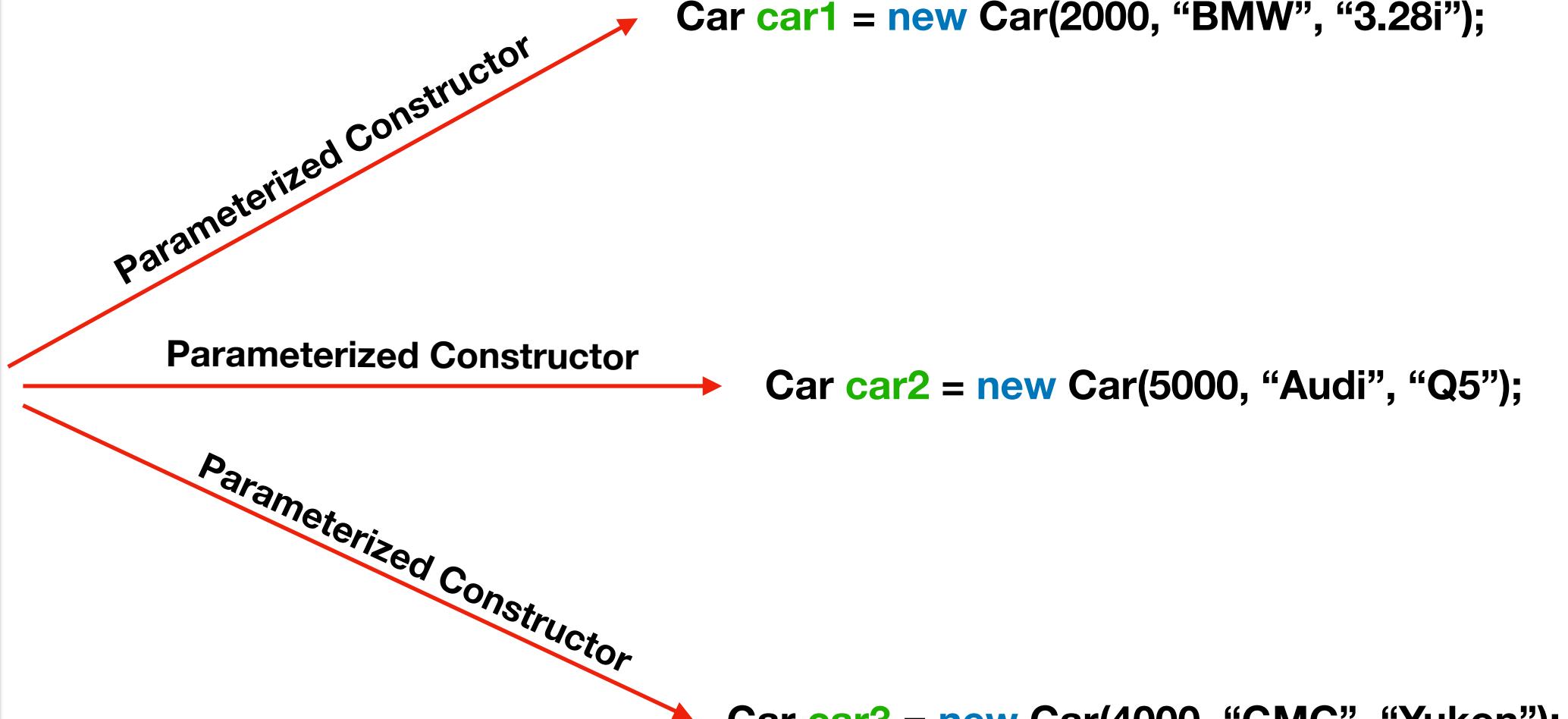
```
car2.price$ = 4000;  
car2.make = "Audi";  
car2.model = "Q5";
```

```
Car car3 = new Car();
```

```
car3.price$ = 3000;  
car3.make = "GMC";  
car3.model = "Yukon";
```

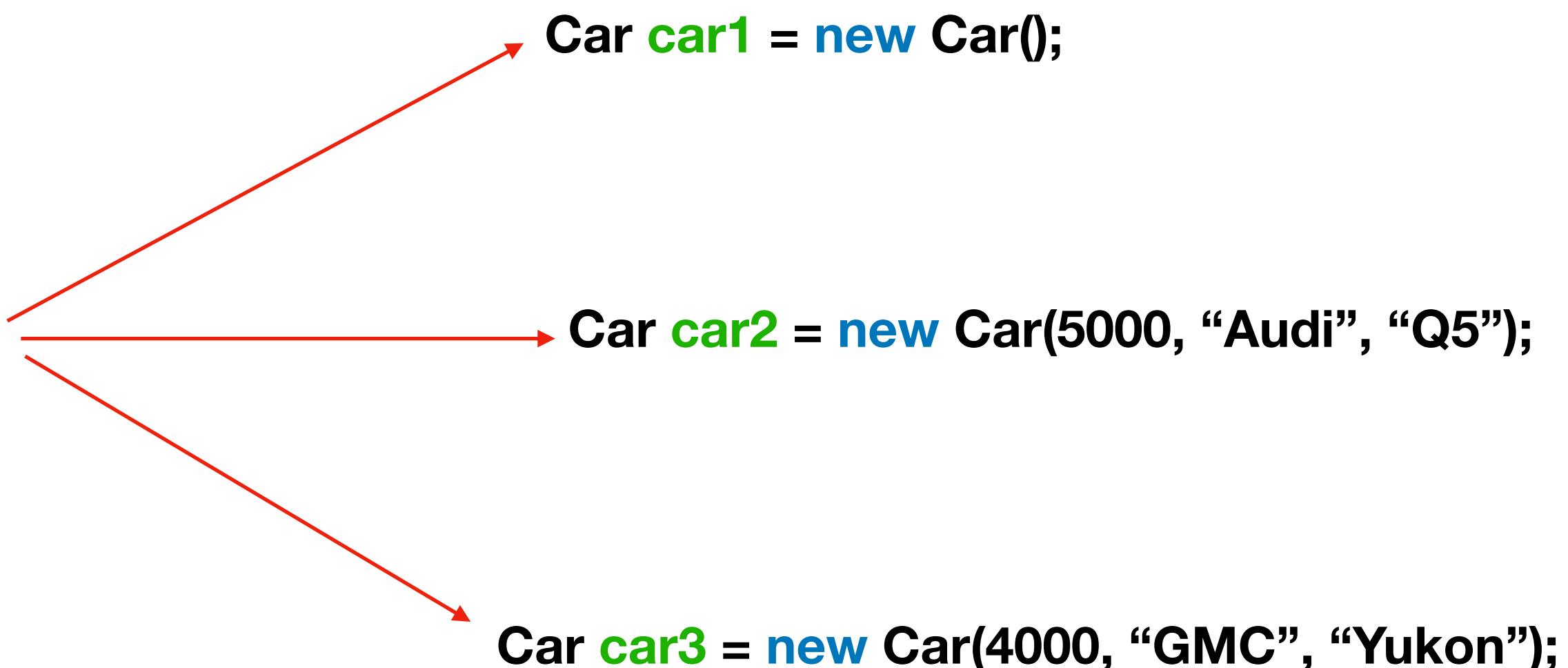


```
public class Car {  
  
    int price$;  
    String make;  
    String model;  
  
    public Car(int price$, String make, String model){  
        this.price$ = price$;  
        this.make = make;  
        this.model = model;  
    }  
}
```



What kind of car does every constructor create?

```
public class Car {  
  
    int price$;  
    String make;  
    String model;  
  
    public Car(int price$, String make, String model){  
        this.price$ = price$;  
        this.make = make;  
        this.model = model;  
    }  
  
}
```



```
Car car1 = new Car();  
Car car2 = new Car(5000, "Audi", "Q5");  
Car car3 = new Car(4000, "GMC", "Yukon");
```



car1, car2 ve car3 hangi constructor'ları kullanır?

```
public class Car {  
  
    int price$;  
    String make;  
    String model;  
  
    public Car(){  
        this.price$ = 2000;  
        this.make = "Honda";  
        this.model = "Accord";  
    }  
  
    public Car(int price$, String make, String model){  
        this.price$ = price$;  
        this.make = make;  
        this.model = model;  
    }  
}
```

Car car1 = new Car();
Car car2 = new Car(5000, "Audi", "Q5");
Car car3 = new Car(4000, "GMC");



Output nedir?

```
public class Student
{
    String name = "Emily";
    int age = 20;

    Student(String name, int age)
    {
        this.name = name;
        this.age = 22;
    }
    public static void main(String[] args)
    {
        Student st = new Student("Oliver", 21);
        System.out.print(st.name);
        System.out.print(", " + st.age);
    }
}
```



output nedir?

```
public class Student {  
    String name;  
    int age;  
    String phone;  
  
    Student() {}  
    Student(String name, int age, String phone) {  
        this.phone = phone;  
        this.name = name;  
    }  
    public static void main(String[] args){  
        Student s1 = new Student();  
        Student s2 = new Student("John", 25, "029-998877");  
        System.out.print(s2.name + ", " + s2.age + ", " + s2.phone)  
    }  
}
```



output nedir?

```
public class MyConstructor {  
    int x = 5;  
    MyConstructor() {  
        System.out.print("-x" + x);  
    }  
    MyConstructor(int x) {  
        this();  
        System.out.print("-x" + x);  
    }  
    public static void main(String[] args){  
        MyConstructor mc1 = new MyConstructor(4);  
        MyConstructor mc2 = new MyConstructor();  
    }  
}
```



Sorular...

```
public class MyClass{  
    int num1;  
    String name = "Ali";  
  
    MyClass(){  
        char letter = 'c';  
    }  
  
    MyClass(int num1){  
        this.num1 = num1;  
    }  
  
    void MyClass(){  
        num1++;  
    }  
  
    increase(int num1){  
        name++;  
    }  
}
```

Boslukları “True” veya “False” ile doldurunuz

- 1) **Turquoises** instance variables dir
- 2) **Orange** olan parametresiz constructordir
- 3) **Pink** olan parametreli constructordir.
- 4) **Green** olan parametreli constructordir.
- 5) **Blue** olan parametreli constructordir.
- 6) Variable **“letter”** local variabledir
- 7) Instance variablelara baslangic degeri atanmalıdır
- 8) Verilen kodda sadece 1 Compile Time Error vardır.
- 9) **“this”** keyword instance variableları ifade eder



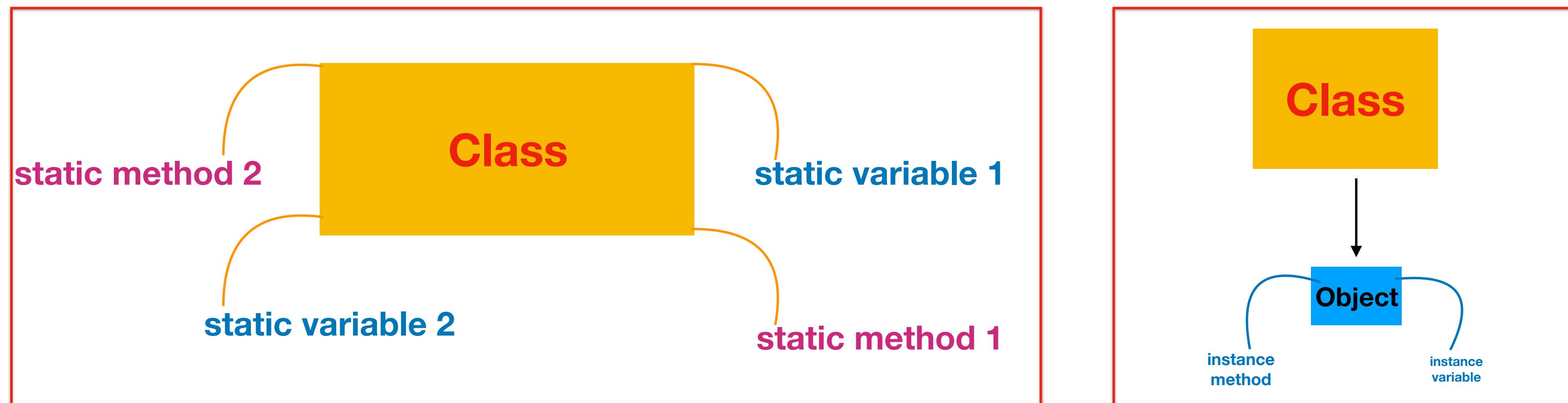
Static Kelimesinin (Keyword) Kullanım Amaci

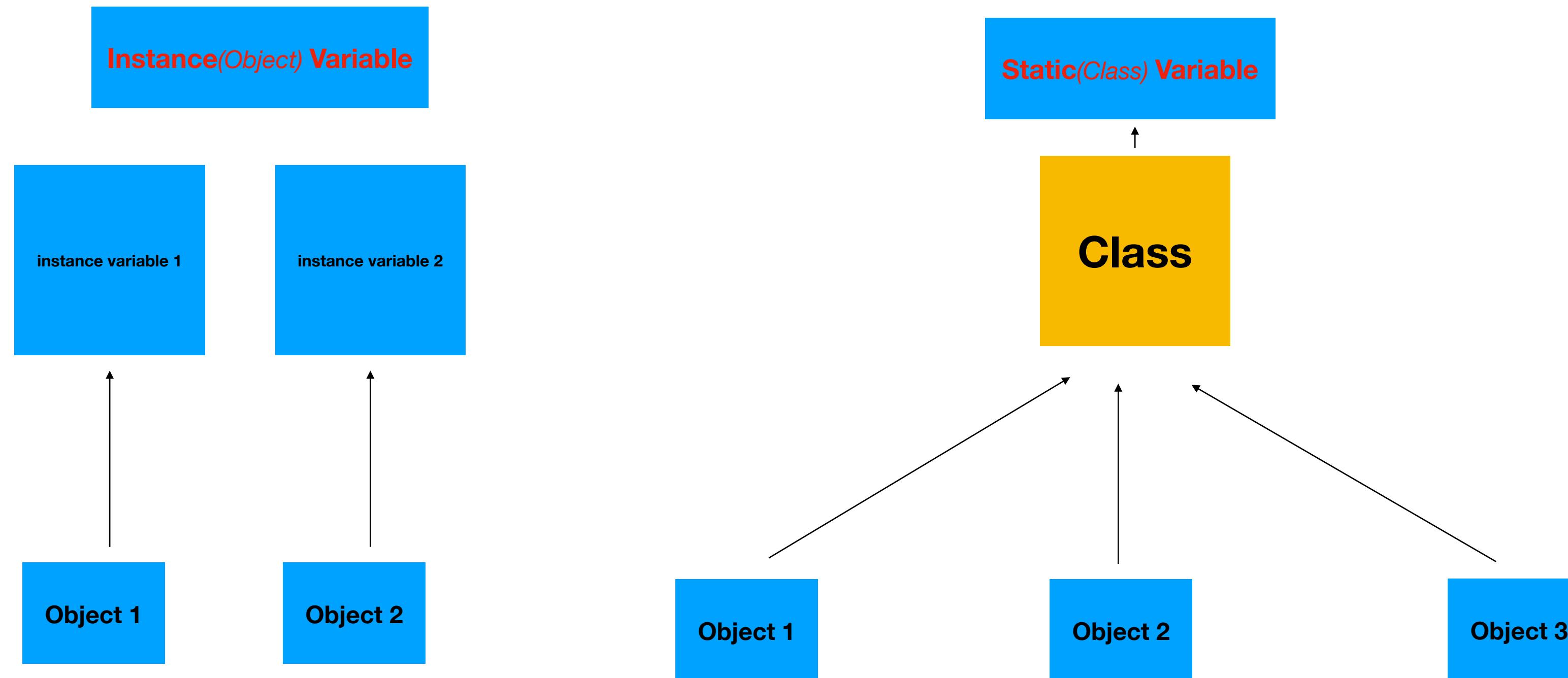
static kelimesi bir **Variable**'i veya **Method**'u **Class**'a bağlamak için kullanılır.

Bir **variable** veya **Method** **static** olarak etiketlendiğinde o artık **class**'in elemanı olmuş demektir.

Bir **variable** veya **method**'u **static** olarak etiketlediğimizde, ona ulaşmak için **object** oluşturmamıza gerek kalmaz.

If you want to go to an instance variable, definitely you have to create an object





- 1) Class yükleniğinde, memory'de static variable'lar oluşturulur.**
- 2) Static variable'lar bir tane oluşturulur ve class'daki tüm objeler onu görür ve kullanır.**
- 3) Memory kullanımı static variable'lar için sadece bir kere olur.**
- 4) Static Variable'lar static methodların içinde kullanılabilir.**
- 5) Static Variable'lara başka classlardan sadece class ismi kullanılarak ulaşılabilir, obje oluşturmaya gerek yok.**

output nedir?

```
class VariableDemo
{
    static int count=0;
    public void increment()
    {
        count++;
    }
    public static void main(String args[])
    {
        VariableDemo obj1=new VariableDemo();
        VariableDemo obj2=new VariableDemo();
        obj1.increment();
        obj2.increment();
        System.out.println("Obj1: count is="+obj1.count);
        System.out.println("Obj2: count is="+obj2.count);
    }
}
```



Output nedir?

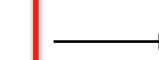
```
public class MyClass
{
    int x;
    static int y;

    MyClass(int i)
    {
        x += i;
        y += i;
    }
    public static void main(String[] args)
    {
        new MyClass(2);
        MyClass mc = new MyClass(3);
        System.out.print(mc.x + "," + mc.y);
    }
}
```



Static Method

```
public static void add(){  
    System.out.println(num1 + num2);  
}
```



Return Type'dan önce **static keyword** kullanarak,
static method oluşturabiliriz



Static Method'lar static variable(class variables) lari kullanabilirler ama static olmayanları object oluşturmadan kullanamazlar

```
class JavaExample{
    static int i = 10;
    static String s = "Beginnersbook";
    //This is a static method
    public static void main(String args[])
    {
        System.out.println("i:"+i);
        System.out.println("s:"+s);
    }
}
```

output nedir?

```
public class Counter {  
    int count;  
    static int stCount;  
  
    public Counter() {  
        count ++ ;  
        stCount ++ ;  
    }  
    public int getCount(){  
        return count;  
    }  
    public static int getStCount(){  
        return stCount;  
    }  
  
    public class TestCounter {  
  
        public static void main(String[] args){  
            Counter cs1 = new Counter();  
            Counter cs2 = new Counter();  
            Counter cs3 = new Counter();  
            Counter cs4 = new Counter();  
            Counter cs5 = new Counter();  
            Counter cs6 = new Counter();  
            System.out.println("count is: " + cs6.getCount());  
            System.out.println("stCount is: " + cs6.getStCount())  
        }  
    }  
}
```



output nedir?

```
public class Counter {  
    int count;  
    static int stCount;  
  
    public Counter() {  
        count ++ ;  
        stCount ++ ;  
    }  
    public int getCount(){  
        return count;  
    }  
    public static int getStCount(){  
        return stCount;  
    }  
}  
  
public class TestCounter {  
  
    public static void main(String[] args){  
        Counter cs1 = new Counter();  
        Counter cs2 = new Counter();  
        Counter cs3 = new Counter();  
        Counter cs4 = new Counter();  
        Counter cs5 = new Counter();  
        Counter cs6 = new Counter();  
        System.out.println("count is: " + cs1.getCount());  
        System.out.println("stCount is: " + cs1.getStCount())  
    }  
}
```



output nedir?

```
public class StaticMember {  
    static int x;  
    int y;  
  
    StaticMember() {  
        x += 2;  
        y ++ ;  
    }  
    static int getSquare(){  
        return x * x;  
    }  
    public static void main(String[] args){  
        StaticMember sm1 = new StaticMember();  
        StaticMember sm2 = new StaticMember();  
        int z = sm1.getSquare();  
        System.out.print("-x" + z + "-y" + sm2.y);  
    }  
}
```



output nedir?

```
class Counter{  
  
    int count=0;  
  
    Counter(){  
        this.count++;  
        System.out.println(count);  
    }  
  
    public static void main(String args[]){  
  
        Counter c1=new Counter();  
        Counter c2=new Counter();  
        Counter c3=new Counter();  
    }  
}
```

output nedir?

```
class Student{  
  
    int number;  
    String name;  
    static String college ="ITS";  
  
    Student(int r, String n, String college){  
        this.number = r;  
        this.name = n;  
        this.college = college;  
    }  
  
    public static void main(String args[]){  
  
        Student s1 = new Student(111,"Karan", "MIT");  
        Student s2 = new Student(222,"Aryan", "Harvard");  
  
        System.out.println(s1.number);  
        System.out.println(s2.number);  
  
        System.out.println(s1.name);  
        System.out.println(s2.name);  
  
        System.out.println(s1.college);  
        System.out.println(s2.college);  
  
    }  
}
```

output nedir?

```
public class MyClass {  
  
    int x = 3;  
    int y = 5;  
  
    MyClass() {  
        x += 1;  
        System.out.print("-x" + x);  
    }  
    MyClass(int i) {  
        this();  
        this.y = i;  
        x += y;  
        System.out.print("-x" + x);  
    }  
    MyClass(int i, int i2) {  
        this();  
        this.x -= 4;  
        System.out.print("-x" + x);  
    }  
    public static void main(String[] args){  
        MyClass mc1 = new MyClass(4,3);  
    }  
}
```



output nedir?

```
class Test1 {  
    static int x = 10;  
public  
    static void main(String[] args)  
    {  
        Test1 t1 = new Test1();  
        Test1 t2 = new Test1();  
  
        t1.x = 20;  
        System.out.print(t1.x + " ");  
        System.out.println(t2.x);  
    }  
}
```



output nedir?

```
class Test1 {  
    int x = 10;  
public static void main(String[] args)  
{  
    Test1 t1 = new Test1();  
    Test1 t2 = new Test1();  
    t1.x = 20;  
    System.out.print(t1.x + " ");  
    System.out.println(t2.x);  
}  
}
```



output nedir?

```
public class Test1 {  
    static int i = 1;  
  
    public static void main(String[] args) {  
  
        for(i=1; i<10; i++) {  
            i = i + 2;  
            System.out.print(i + " ");  
        }  
        System.out.println("==> " + i);  
  
    }  
}
```

