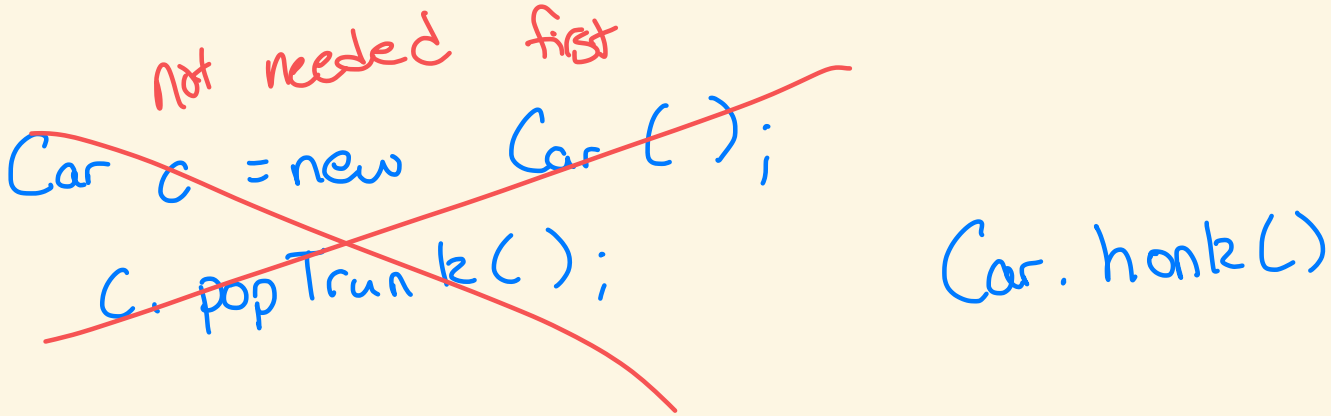


STATIC VS NON-STATIC METHODS

- Also called "class methods"
- Invoked through the class name
- Don't need to have object instantiated
- Example: `sqrt` method in `Math` class

Math.sqrt(4);

STATIC METHODS

- add `static` modifier in method declaration
- cannot access instance variables
- can reference static variables

private static int count = 0;

private String color;

keep track how many cars made

public Car ( ) {

count += 1;

}

STATIC VARIABLES

Other variable types:

- local variable: variables declared inside a method (disappear after method ends)
- instance variable: each instance of class has it's own copy

private int day;

Static variable:

- shared among all instances of the class

STATIC VARIABLES

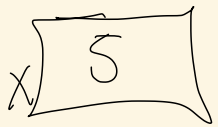
- use `static` modifier
- also called "class variables"
- changing it changes for all
- local variables cannot be static
- constants (declared with `final` are also often `static`)

"THIS" REFERENCE

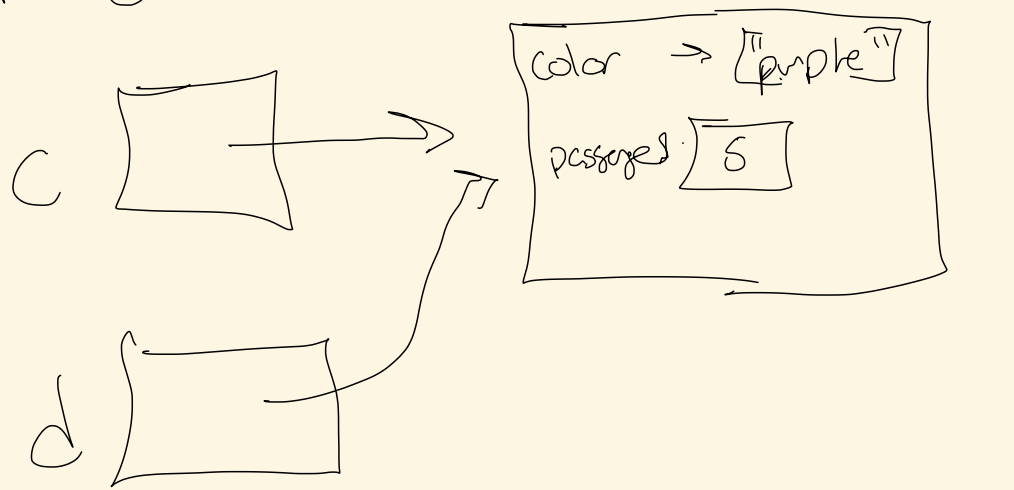
- `this` is a reference to object through which method was invoked
- often used when coonstructors have parameters with same name as instance variables

```
public Class Car {
private String color;
private int passengers;

public Car (String color, int passengers) {
    this.passengers = passengers;

}
public void resize () {
    passengers--;
}
```

```
int x = 5;
```




```
Car c = new Car("blue", 5);
```




```
Car d = c;
d.setColor("purple");
```

```
Car c = new Car("yellow", 4);
Car d = new Car("black", 8);
myster(d, c);

public void myster(Car c, Car d) {
    Car q = c;
    c.setColor(d.getColor());
    q.setPassengers(7);
    d.setPassengers(3);
    Car p = new Car("orange", 2);
    p = d
    p.setColor("red");
```