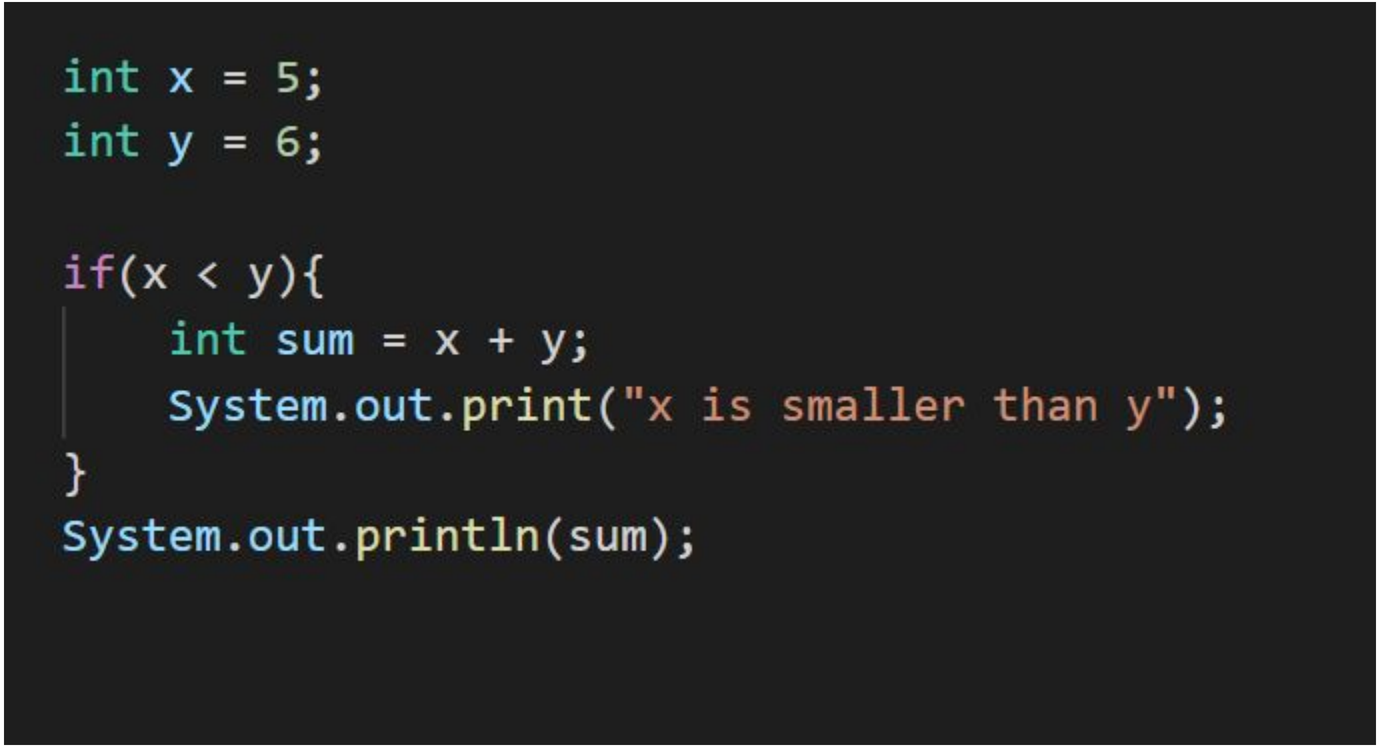
Global Variables & Constructors

Java
Mr. Poole

Scope of Variables

What would happen here?



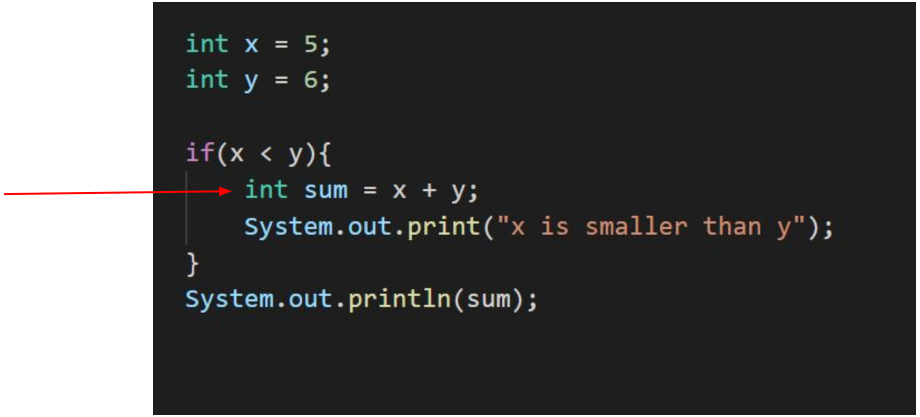
```java
int x = 5;
int y = 6;

if(x < y){
    int sum = x + y;
    System.out.print("x is smaller than y");
}
System.out.println(sum);
```

What would happen here?

It won’t compile!

Variables constructed within brackets (squigglys)
are only usable within the same brackets

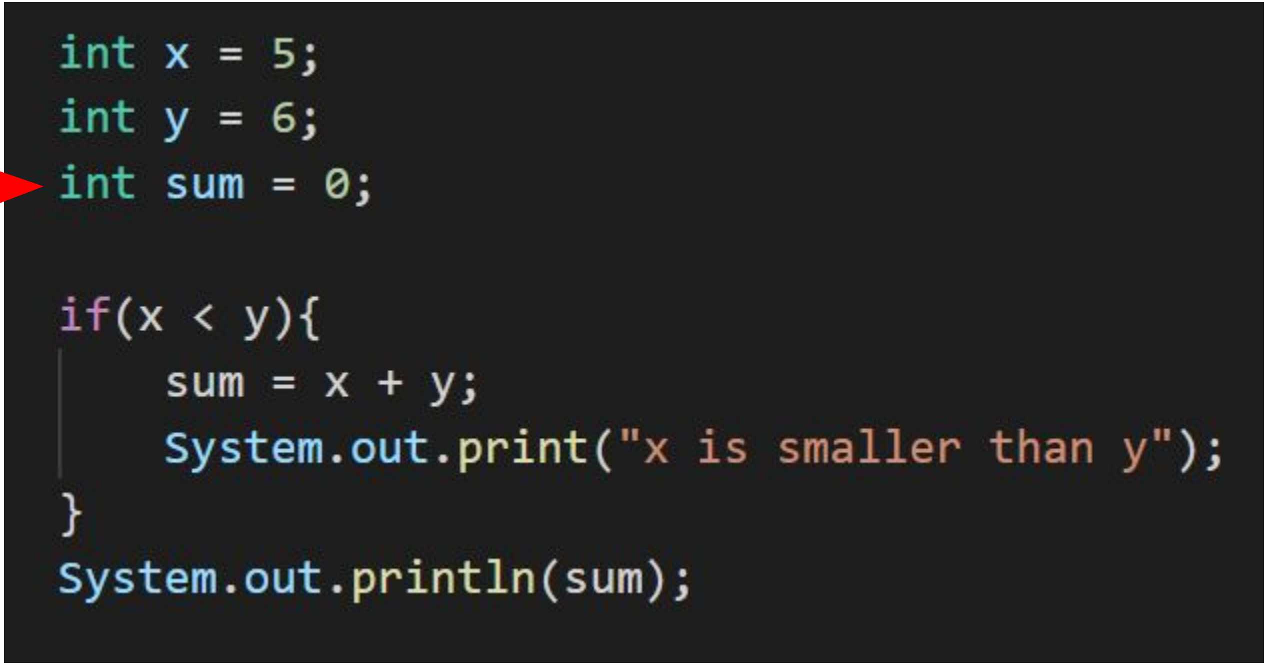
Since sum is constructed within the **if** statement, sum can’t be accessed outside.



```java
int x = 5;
int y = 6;

if(x < y){
    int sum = x + y;
    System.out.print("x is smaller than y");
}
System.out.println(sum);
```

What would happen here?

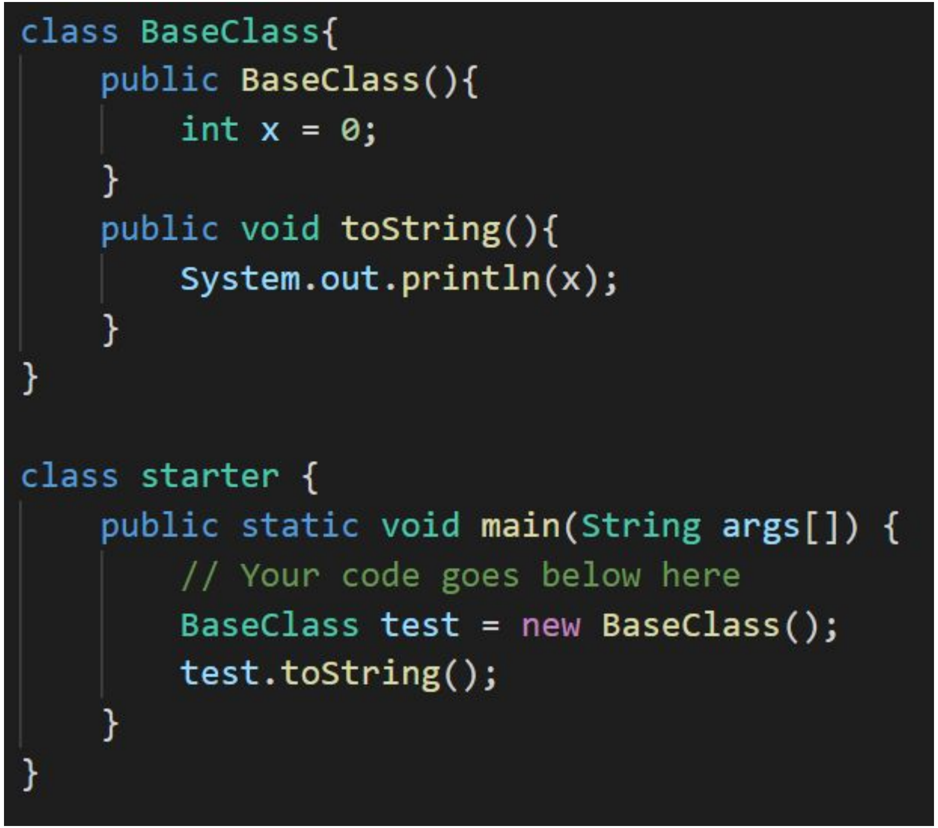
Construct sum outside of the if statement and access it within.
This is an example of variables being used “globally”



```java
int x = 5;
int y = 6;
int sum = 0;

if(x < y){
    sum = x + y;
    System.out.print("x is smaller than y");
}
System.out.println(sum);
```

Let’s apply this to Classes/Methods!

Would this compile?



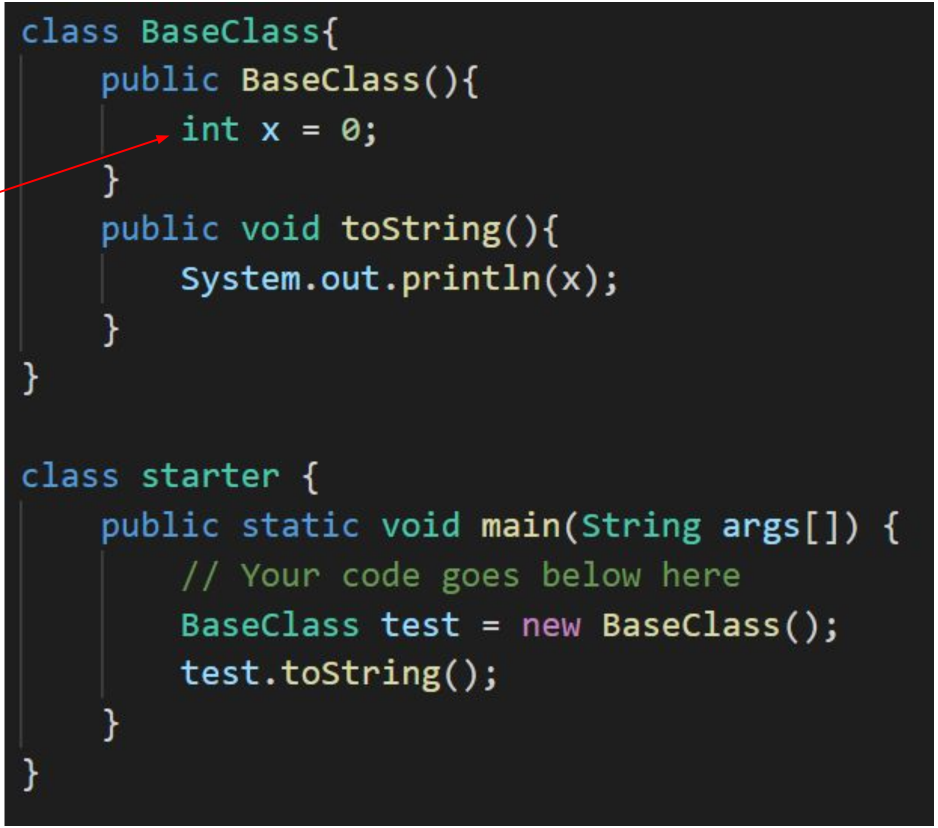
```java
class BaseClass{
    public BaseClass(){
        int x = 0;
    }
    public void toString(){
        System.out.println(x);
    }
}

class starter {
    public static void main(String args[]) {
        // Your code goes below here
        BaseClass test = new BaseClass();
        test.toString();
    }
}
```

Let’s apply this to Classes/Methods!

It wouldn’t!

`int x` is constructed in the **constructor**, the `toString` method can’t access x.

To fix this, `int x` will be a global variable!



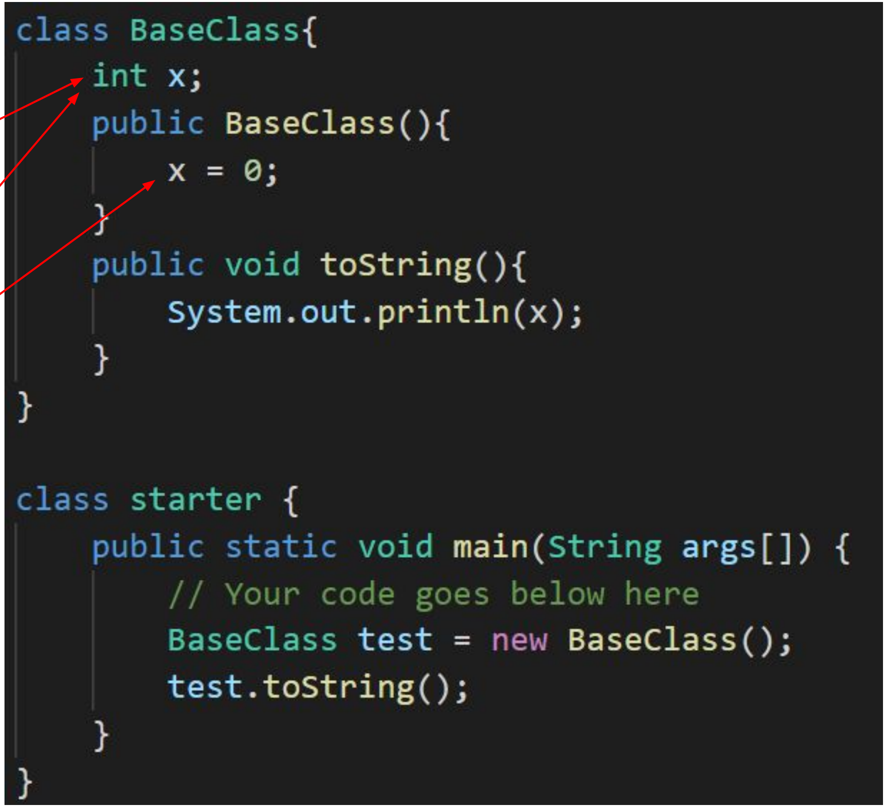
```
class BaseClass{
    public BaseClass(){
        int x = 0;
    }
    public void toString(){
        System.out.println(x);
    }
}

class starter {
    public static void main(String args[]) {
        // Your code goes below here
        BaseClass test = new BaseClass();
        test.toString();
    }
}
```

Let’s apply this to Classes/Methods!

`int x` is now a global variable!

`int x` is constructed in the class
But declared in the constructor.

This means `toString`
can now use x.



```java
class BaseClass{
    int x;
    public BaseClass(){
        x = 0;
    }
    public void toString(){
        System.out.println(x);
    }
}

class starter {
    public static void main(String args[]) {
        // Your code goes below here
        BaseClass test = new BaseClass();
        test.toString();
    }
}
```

Global Variables

These are attributes that the class should have!

In this case, they are the different features of cars! This would be a list of features that EVERY car has.

Global Variables

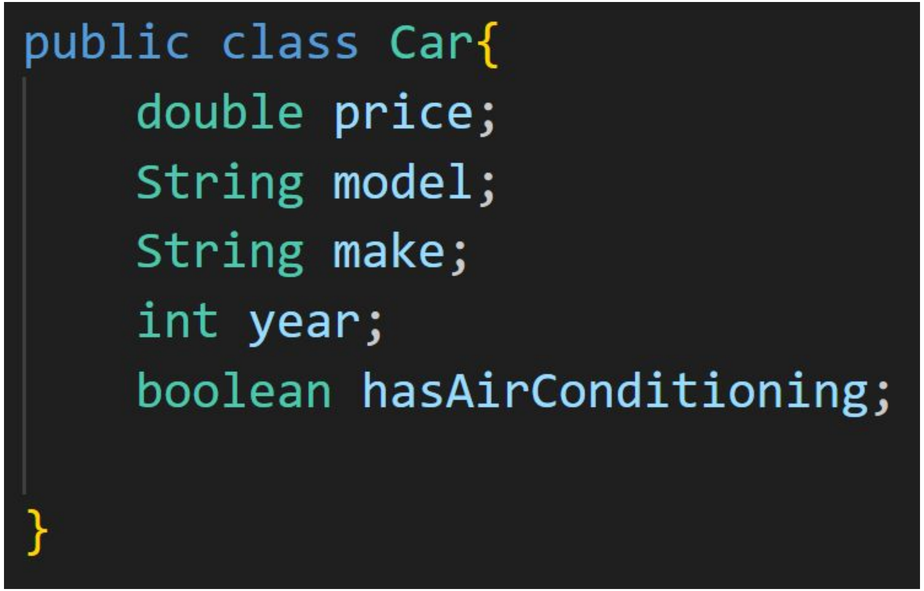
Example Variables

Class	Global Variables
Car	double price String model String make int year boolean hasAirConditioning

Global Variables in Classes

Global variables are **ONLY constructed** inside the class usually at the start of the file.

Being made inside the class allows their scope to be the entire class.

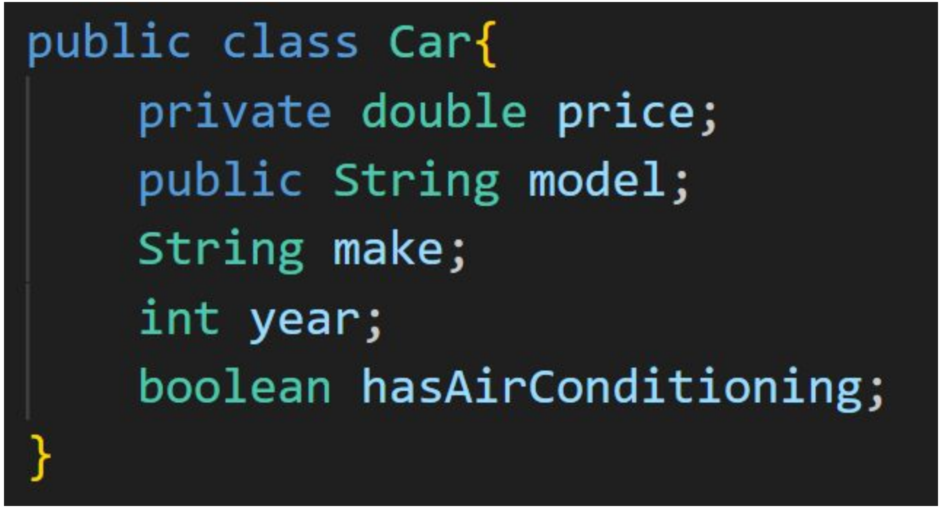


```
public class Car{
    double price;
    String model;
    String make;
    int year;
    boolean hasAirConditioning;

}
```

Access Modifying Global Variables

Global variables also can have access modifiers!

Notating these modifiers allows/doesn’t allow users to directly access the variable values with edit/read access.



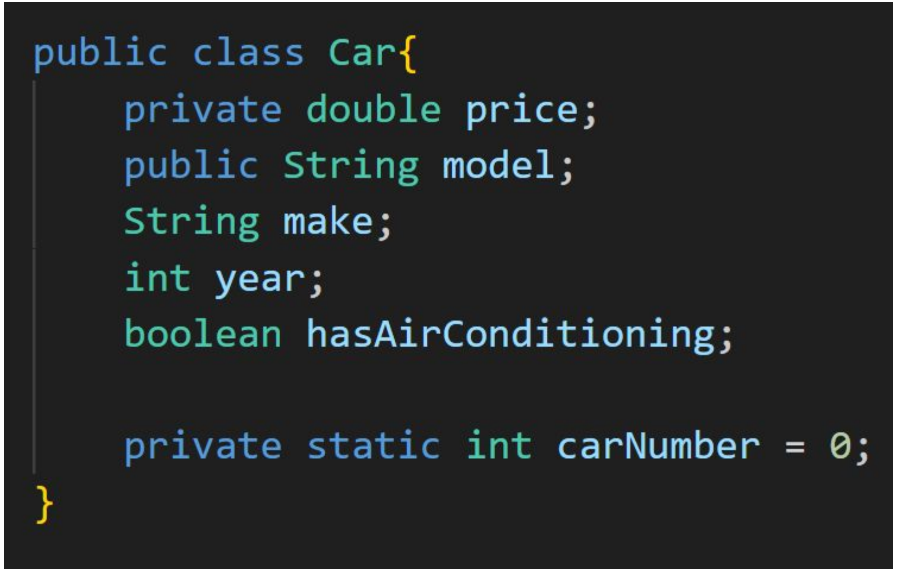
```
public class Car{
    private double price;
    public String model;
    String make;
    int year;
    boolean hasAirConditioning;
}
```

Static Global Variables

Static variables DO NOT change across ALL objects.

In this case, carNumber would be the same and change for ALL Car objects when it is changed.

carNumber can represent how many total cars exist.



```
public class Car{
    private double price;
    public String model;
    String make;
    int year;
    boolean hasAirConditioning;

    private static int carNumber = 0;
}
```

Constructors

Constructor

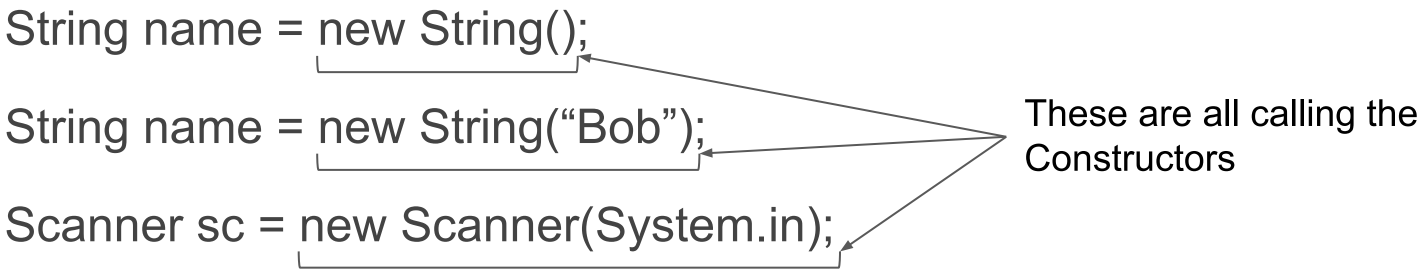
When the object is created, we should set it up for success!

Constructors run at creation and **set up ALL global variables**.

What is a Constructor?

A Constructor is similar to a **Method** but instead it **creates** the object itself.

Example of Constructors that we’ve used.



```
String name = new String();

String name = new String(“Bob”);

Scanner sc = new Scanner(System.in);
```

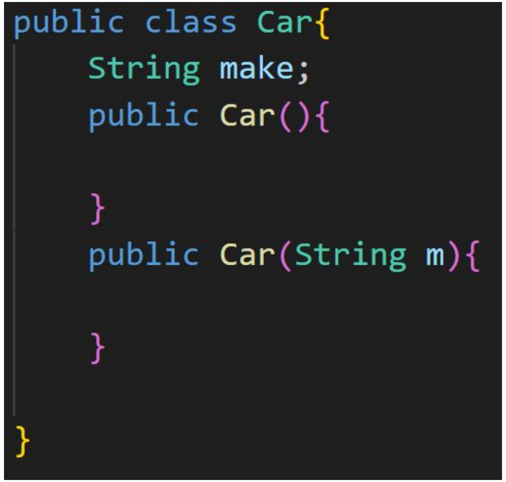
These are all calling the Constructors

Constructors

Constructors can be empty **or** they can be filled of parameters.

Car x = new Car(); Empty Constructor

Car y = new Car(“Toyota”); String Constructor



```
public class Car{
    String make;
    public Car(){

    }
    public Car(String m){

    }

}
```

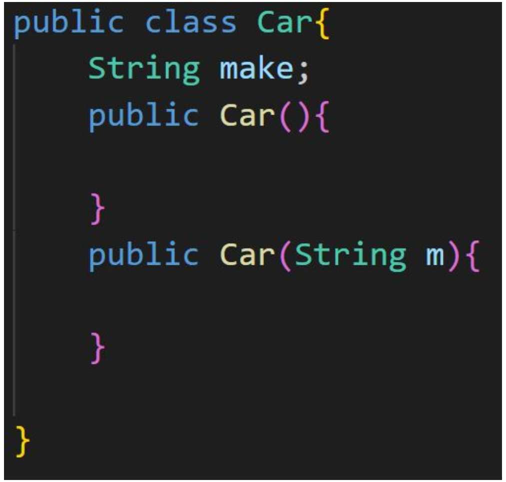
Empty Constructor

String Constructor

Example: Constructors

Constructor’s purpose is to **create** the object.

In this case, it is CREATING the Car with or without a String parameter.



```
public class Car{
    String make;
    public Car(){

    }
    public Car(String m){

    }

}
```

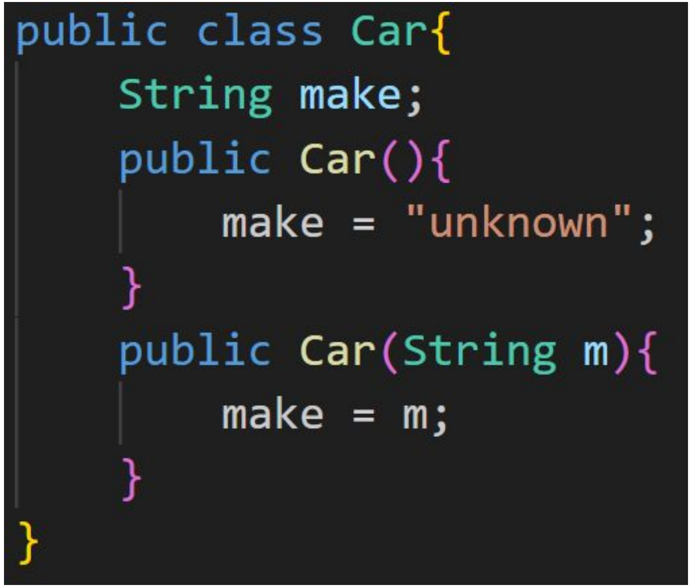
Empty Constructor

String Constructor

Constructors Good Practice

Constructors are great for initializing/declaring values for our global variables!

Variable is constructed below, but declared inside the constructor.



```
public class Car{
    String make;
    public Car(){
        make = "unknown";
    }
    public Car(String m){
        make = m;
    }
}
```

Lab - Global Variables & Constructors

Create the BankAccount Class.

- All bank accounts should have the following information
  - Owner, Account Number, Balance, and isActive
  - nextAccountNumber should be the same for all accounts
    - It starts as a random account value from 50-1000
  - All information should be private!
- Make 3 constructors
  - Default constructor when a bank account opens up

Lab - Global Variables & Constructors

- Make 3 constructors
    - Default constructor when a bank account opens up
        - Values should be set to unknown or 0.
        - Accounts created are active
        - The account number should be set to the next available number
    - The Owner constructor
        - Set up the owner with an active account and a balance of 0
    - The Owner and Initial Deposit constructor
        - Set up the owner with their active account and initial balance
- Make sure when you create a bank account and assign a number, that account number should no longer be available.

Lab - Global Variables & Constructors

Starter.java

In your main, do the following:

- Create 4 bank accounts with various users and constructors!