

Java Classes

Primitive variables vs. Object variables

Primitive Variables

- Primitive variables abstractly represent a memory location to store the data.
- Once a primitive variable is declared, that space is available for use in the program.
- Examples:

```
int homeArea = 2500;
int counter;
```

homeArea	2500
counter	0

It's better to assign an initial value to a variable than assume it has 0.

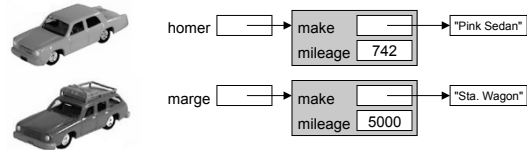
Object Variables

- Object variables abstractly represent a memory location that stores the location of the object.
 - We say that an object variable holds a reference (or a pointer) to the actual object.
- Once an object variable is declared, it must either be:
 - initialized to reference a new object
 - or
 - initialized to an object that has already been created

Object Variables

- Examples:

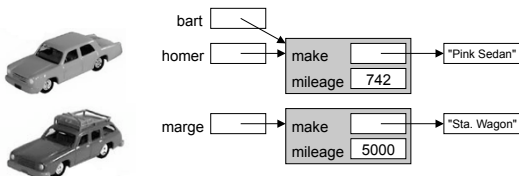
```
Car homer = new Car("Pink Sedan", 742);
Car marge = new Car("Sta. Wagon");
```



Object Variables (cont'd)

- Example:

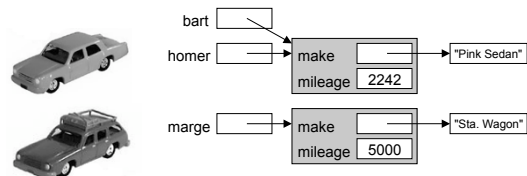
```
Car bart = homer;
```



Object Variables (cont'd)

- Bart goes for a joy ride to Knoxville, Tennessee:

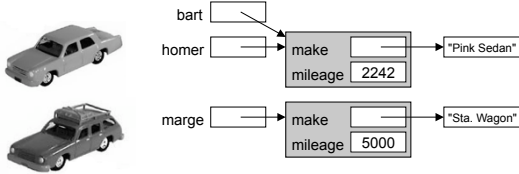
```
bart.drive(1500);
```



Object Variables (cont'd)

- Homer checks his odometer the next day:

```
System.out.println(homer.getMileage());
```

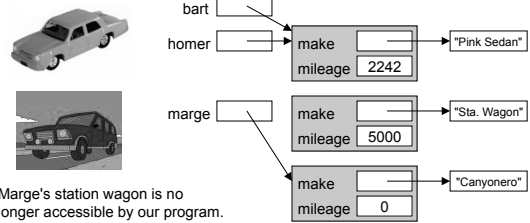


7

Object Variables (cont'd)

- Marge trades in her station wagon for an SUV:

```
marge = new Car("Canyonero", 0);
```

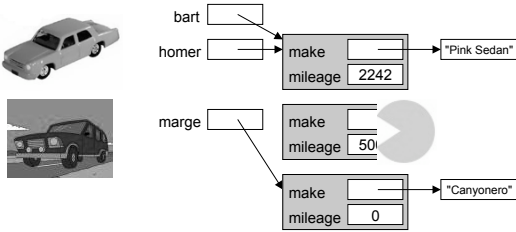


Marge's station wagon is no longer accessible by our program.

8

Garbage Collection

- Java has a garbage collector that (eventually) reclaims any memory that has objects that can't be referenced anymore.

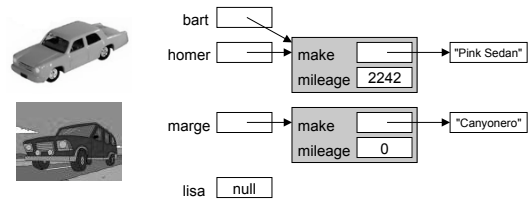


9

The null reference

- An object variable holds the special value **null** if it isn't initialized.

```
Car lisa;
```

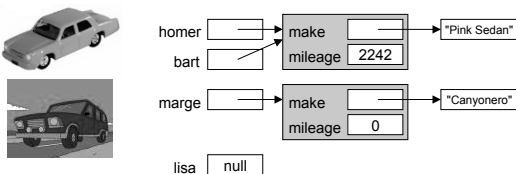


10

NullPointerException

- If we try to call a method using a **null** object reference, we will get **NullPointerException** during runtime.

```
lisa.drive(402);
```



11

Summary

- We can use a primitive variable in our program statements as soon as it is declared.
- We can use an object variable in our program safely as long as it has been initialized to reference a new object or an object that has already been created.
- Memory that belongs to objects that are no longer referenced is (eventually) reallocated back to the computer system for other uses.
 - If not, we get a "memory leak".
- Using a null reference to access an object's methods will result in a runtime error.

12