

Lecture 9: MPG Symbol table

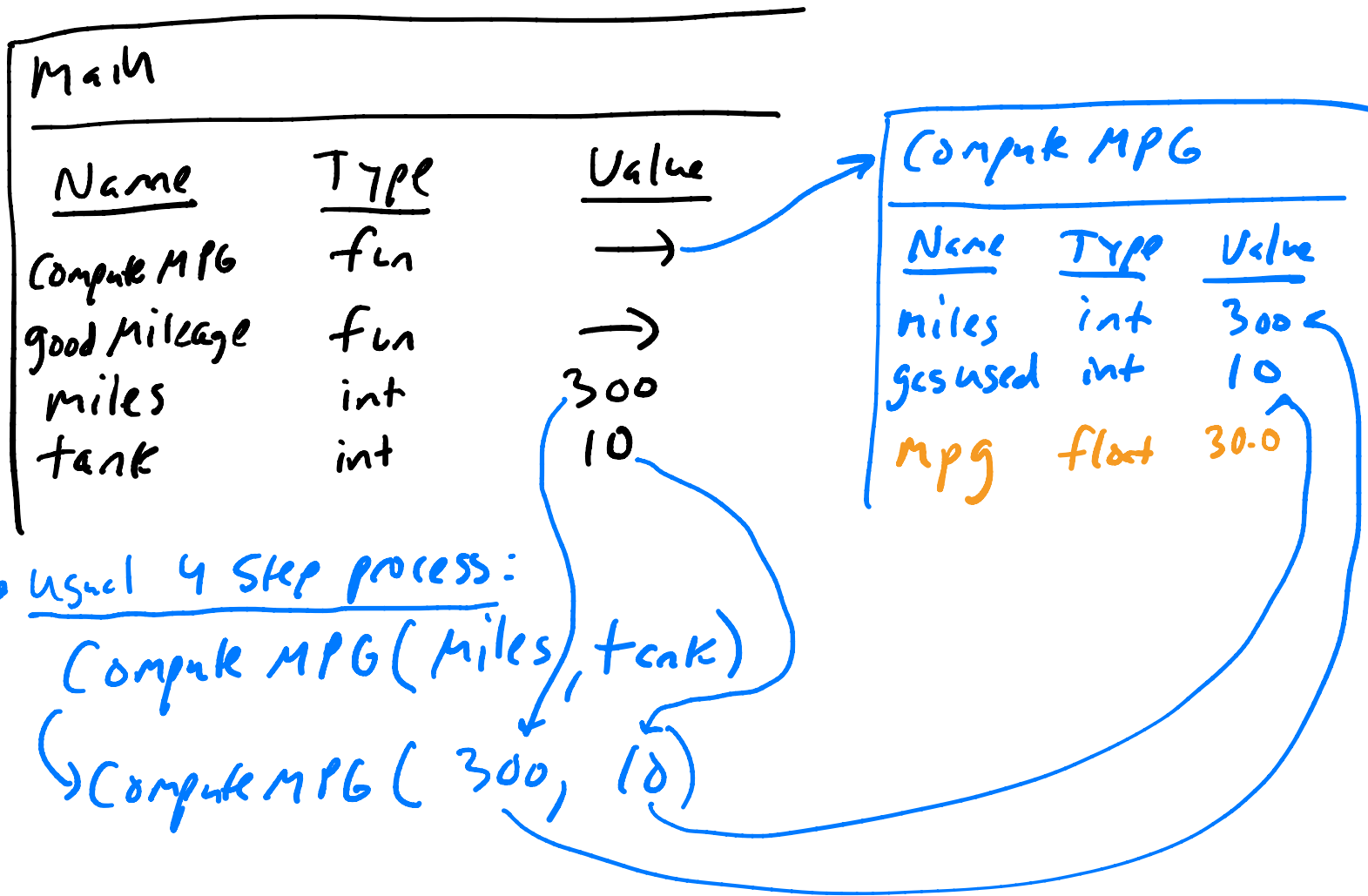
Compute MPG Symbol Table

```
1 def computeMPG(miles, gasUsed):
2     '''Calculate miles-per-gallon, given miles traveled and gallons used'''
3     mpg = miles / gasUsed
4     return mpg
5
6 def goodMileage(mpg):
7     '''Report how good the gas mileage as miles-per-gallon is'''
8     if mpg > 50:
9         print(mpg, 'mpg is great gas mileage!')
10    elif mpg >= 30:
11        print(mpg, 'mpg is pretty good gas mileage.')
12    elif mpg >= 20:
13        print(mpg, 'mpg is acceptable.')
14    else:
15        print('The gas mileage is not great!')
16
17 # main code
18 # Calculate the MPG of our trip and determine how good it is
19 miles = 300
20 tank = 10
21 myMPG = computeMPG(miles, tank)
22 goodMileage(myMPG)
```

After line 20 done:

Main		
<u>Name</u>	<u>Type</u>	<u>Value</u>
computeMPG	fun	→
goodMileage	fun	→
miles	int	300
tank	int	10

line 21:



line 3: Add mpg to function table

line 4: Return statement.

1) Replace Variable with Value (if there is a Variable)
return mpg → return 30.0

2) Replace function call in Main with return
Value copied from function table.

21: myMPG = Compute MPG(300, 10)



myMPG = 30.0

3) Do variable assignment. Add variable myMPG to main symbol table, give it the type/value on the right side of assignment

Main		
<u>Name</u>	<u>Type</u>	<u>Value</u>
Compute MPG	fun	→
good Mileage	fun	→
miles	int	300
-tank	int	10
myMPG	float	30.0

Compute MPG		
<u>Name</u>	<u>Type</u>	<u>Value</u>
miles	int	300
gas used	int	10
mpg	float	30.0

4) Function table goes away

22: Create function symbol table (usual 4 step process) for goodMileage.

Main		
<u>Name</u>	<u>Type</u>	<u>Value</u>
compute MPG	fun	→
good mileage	fun	→
miles	int	300
-tank	int	10
my MPG	float	30.0

good mileage		
<u>Name</u>	<u>Type</u>	<u>Value</u>
mpg	float	30.0

- Evaluate Conditional, elif $mpg \geq 30$ is true, print on line 11, function ends, good mileage symbol table goes away, program ends.