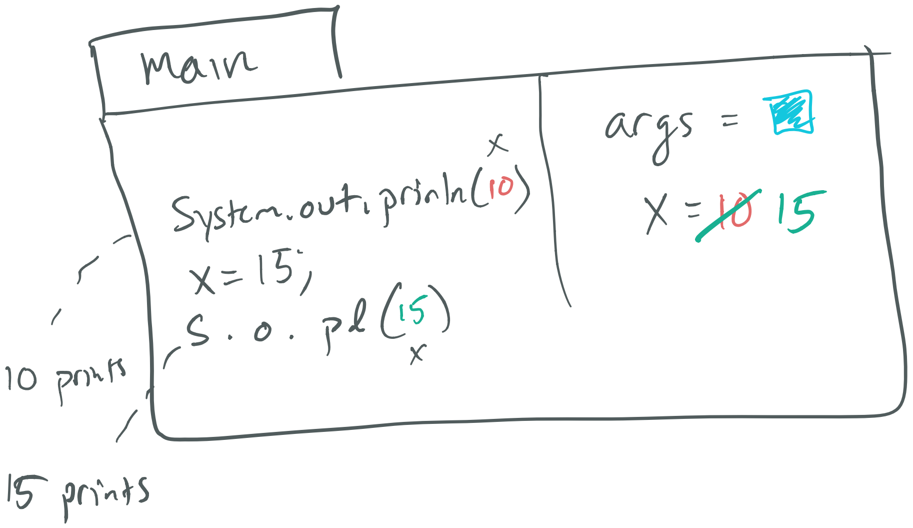
(Local) Variables + Variable Update

Inside a method:

```
int x = 10;
```

$$\underbrace{\text{String}}_{\text{type}}\ \underbrace{s}_{\text{name}} = \underbrace{\text{"abc" + "def"}}_{\text{expression}};$$

Variable definition
declaration



```
x = 15;
```

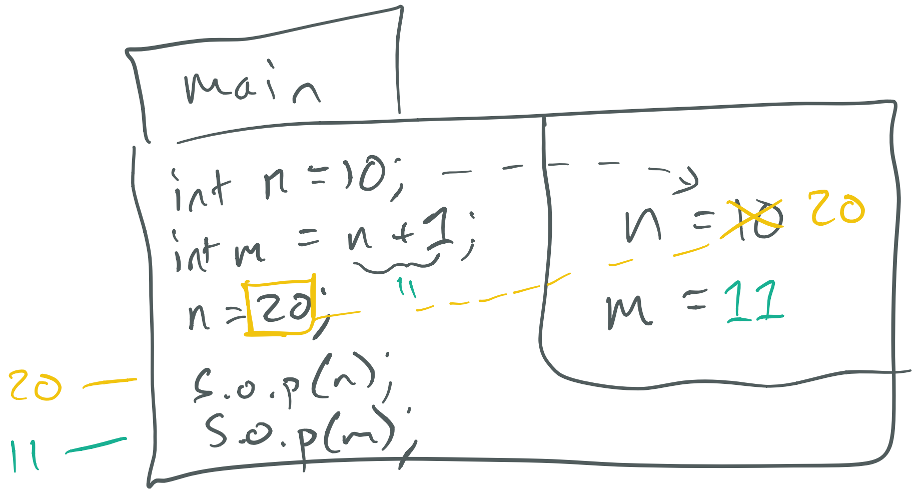
Variable assignment
update

statements

<name> = <expression>;

$$\underbrace{x = \overbrace{x + 1}^{\text{this evaluates to a value,}};}_{\text{This makes little sense in math notation}}$$

then updates value stored in x



```
int n = 10;
int m = n + 1;
n = 20;
S.o.p(n);
S.o.p(m);
```

(In Java)

A variable update can only affect the value stored within a single variable.

Loops – doing more useful work with arrays

String:

int   length()

takes no arguments, returns # of characters in Strg

Arrays

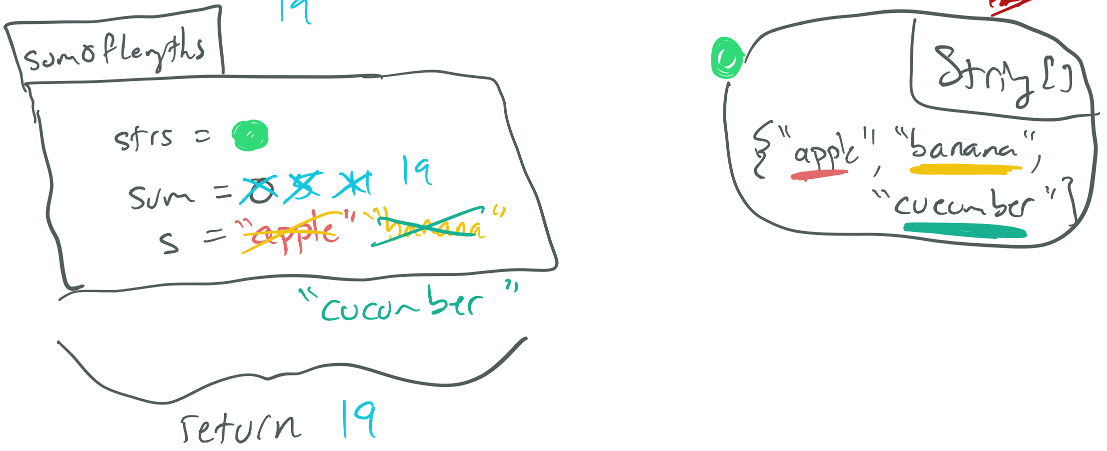
.length – special length field of Arrays,
arrays have no .length() method

```
int sum=0;
for (String s: strs) {
    sum = sum + s.length();
}
return sum;
```

for each element in this array, evaluate loop with that element in s

0 + 5
5
5 + 6
11
11 + 8
19

Top right box should be Strng[3]!



sumOfLengths
strs =
sum = ~~0~~ ~~5~~ ~~11~~ 19
s = ~~"apple"~~ ~~"banana"~~ "cucumber"
return 19
String[]
{"apple", "banana", "cucumber"}

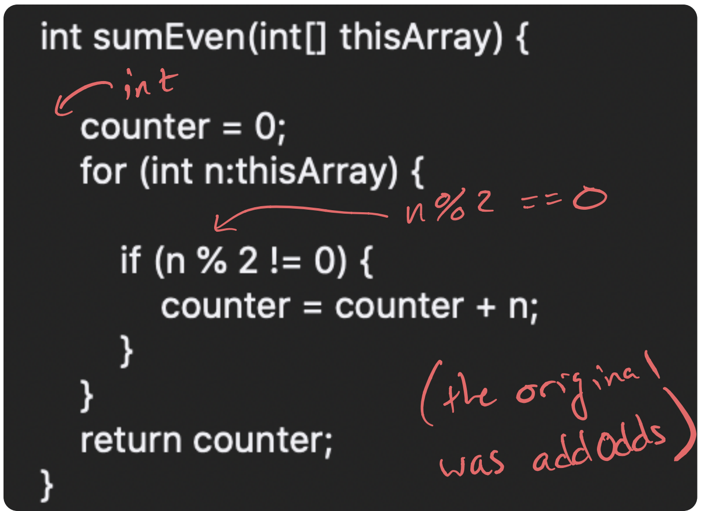
enhanced for loops
for-in
for-each

<type> must be the type of elements in <array>

```
for (<type> <name>: <array>) {
    <body>
}
```

{1, 2, 7, -1}

counter	n	n%2
~~0~~	1	1
1	2	0
1		

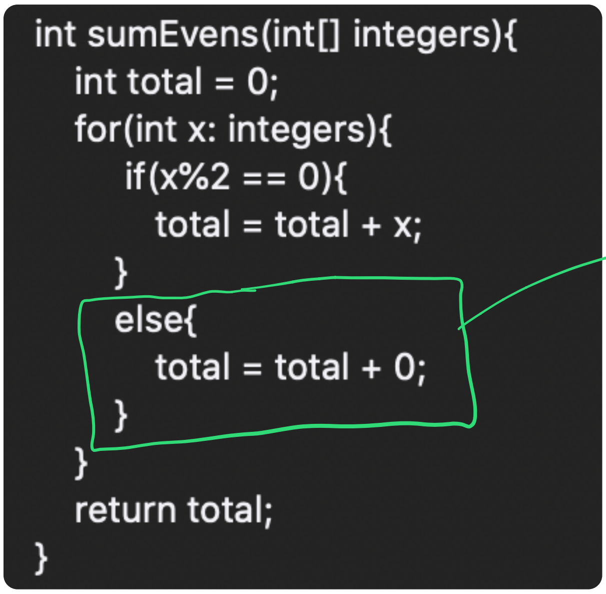


```
int sumEven(int[] thisArray) {
  counter = 0;
  for (int n:thisArray) {
    if (n % 2 != 0) {
      counter = counter + n;
    }
  }
  return counter;
}
```

int

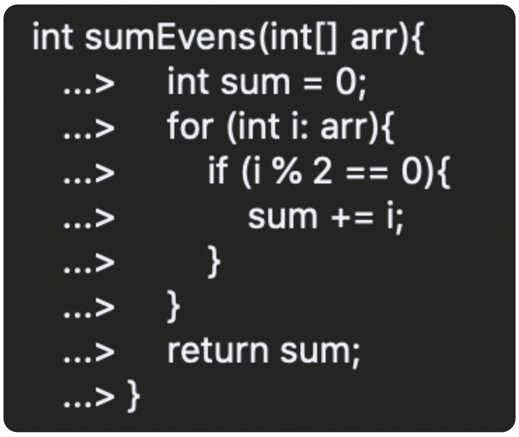
n%2 == 0

(the original was addOdds)



```
int sumEvens(int[] integers){
    int total = 0;
    for(int x: integers){
        if(x%2 == 0){
            total = total + x;
        }
        else{
            total = total + 0;
        }
    }
    return total;
}
```

Doesn't change any answers



```
int sumEvens(int[] arr){
   ...>     int sum = 0;
   ...>     for (int i: arr){
   ...>         if (i % 2 == 0){
   ...>             sum += i;
   ...>         }
   ...>     }
   ...>     return sum;
   ...> }
```

↑ we should ignore this part as human readers