Objects

Discussion 4

Announcements

- HW 2 due Tuesday 09/14
- Lab - Proj 0 OH! (see Piazza)
- Proj 0 due Friday 09/17

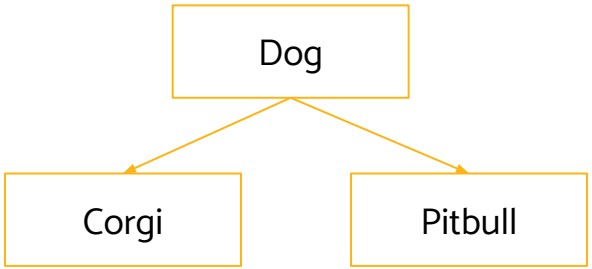
Review

Classes

Subclasses (or child classes) are classes that extend another class. This means that they have access to all of the functions and variables of their parent class in addition to any functions and variables defined in the child class.

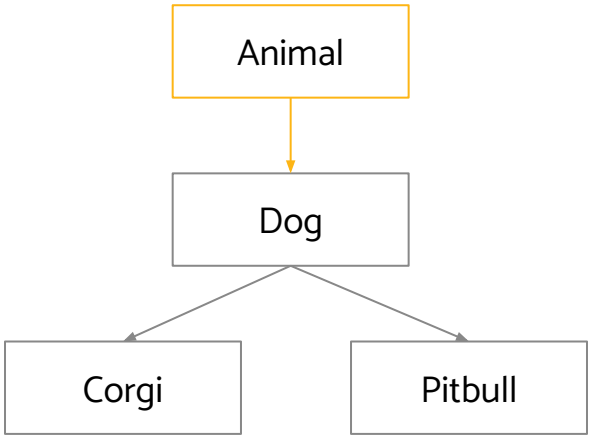
Superclasses or parent classes are classes that are extended by another class. You can use `super(...)` to call the parent constructor of a subclass.

Classes can only extend one class but can be extended by many classes.



Abstract Classes

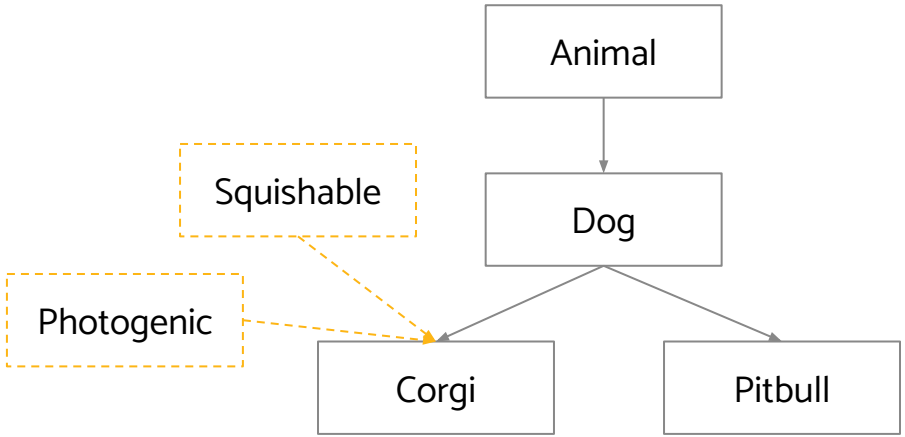
Abstract Classes are classes that cannot be directly referenced. Instead, they must be extended by a **concrete class**. They describe all of the functions that a class of their "type" must be able to do, with or without implementation.



Interfaces

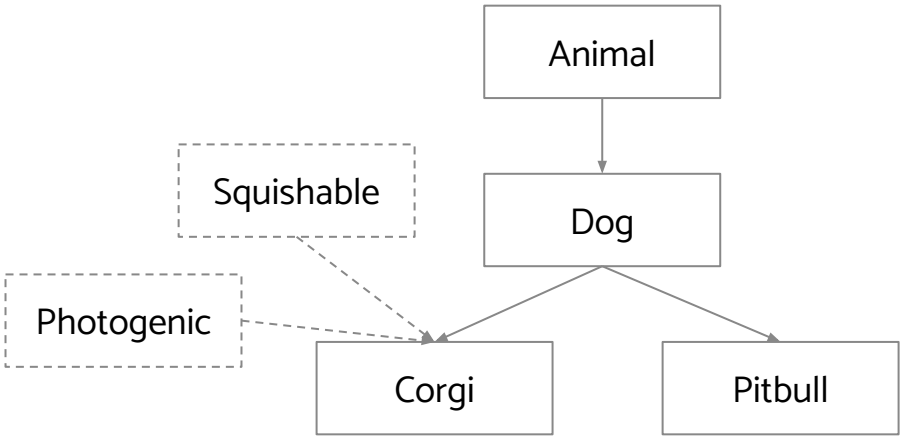
Interfaces are implemented by classes. They describe a narrow ability that can apply to many classes that may or may not be related to one another. They are like abstract classes in that they do not usually implement the methods they specify. One class can implement many interfaces.

Ex. Comparable, List



Implementation

```
abstract class Animal {...}

interface Squishable {...}

interface Photogenic {...}

class Dog extends Animal {...}

class Pitbull extends Dog {...}

class Corgi extends Dog implements Squishable, Photogenic {...}
```



Worksheet

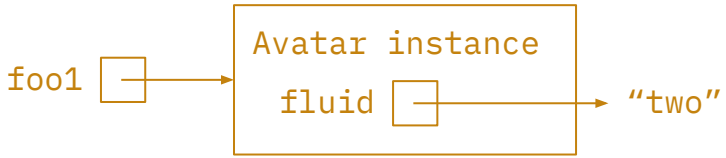
1A Objects Review

```
1  public class Avatar {
2      public static String electricity;
3      public String fluid;
4
5      public Avatar(String str1, String str2) {
6          Avatar.electricity = str1;
7          this.fluid = str2;
8      }
9
10     public static void main(String[] args) {
11         Avatar foo1 = new Avatar("one", "two");
12         Avatar foo2 = new Avatar("three", "four");
13         System.out.println(foo1.electricity + foo1.fluid);
14         foo1.electricity = "I declare ";
15         foo1.fluid = "a thumb war";
16         System.out.println(foo2.electricity + foo2.fluid);
17     }
18 }
```

What would be printed after executing the main method?

1A Objects Review

```
1   public class Avatar {
2       public static String electricity;
3       public String fluid;
4
5       public Avatar(String str1, String str2) {
6           Avatar.electricity = str1;
7           this.fluid = str2;
8       }
9
10      public static void main(String[] args) {
11          Avatar foo1 = new Avatar("one", "two");
12          Avatar foo2 = new Avatar("three", "four");
13          System.out.println(foo1.electricity + foo1.fluid);
14          foo1.electricity = "I declare ";
15          foo1.fluid = "a thumb war";
16          System.out.println(foo2.electricity + foo2.fluid);
17      }
18  }
```

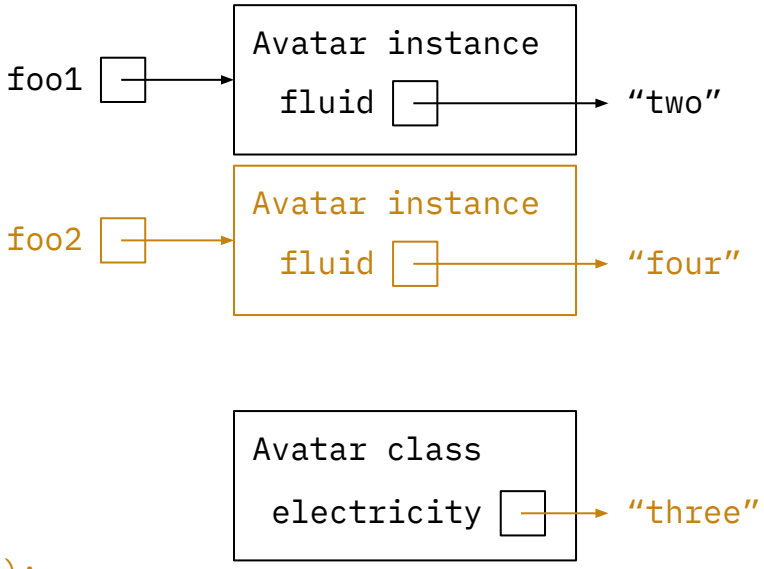


Console:

What would be printed after executing the main method?

1A Objects Review

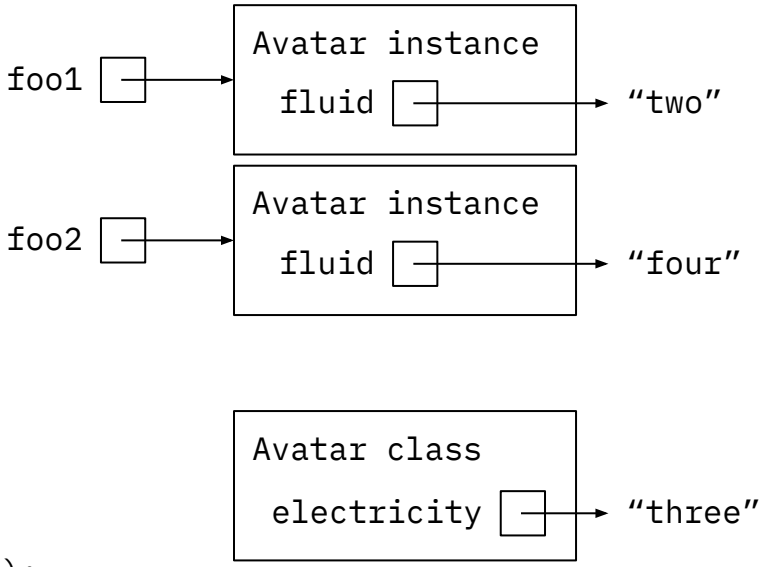
```
1   public class Avatar {
2       public static String electricity;
3       public String fluid;
4
5       public Avatar(String str1, String str2) {
6           Avatar.electricity = str1;
7           this.fluid = str2;
8       }
9
10      public static void main(String[] args) {
11          Avatar foo1 = new Avatar("one", "two");
12          Avatar foo2 = new Avatar("three", "four");
13          System.out.println(foo1.electricity + foo1.fluid);
14          foo1.electricity = "I declare ";
15          foo1.fluid = "a thumb war";
16          System.out.println(foo2.electricity + foo2.fluid);
17      }
18  }
```



What would be printed after executing the main method?

1A Objects Review

```
1   public class Avatar {
2       public static String electricity;
3       public String fluid;
4
5       public Avatar(String str1, String str2) {
6           Avatar.electricity = str1;
7           this.fluid = str2;
8       }
9
10      public static void main(String[] args) {
11          Avatar foo1 = new Avatar("one", "two");
12          Avatar foo2 = new Avatar("three", "four");
13          System.out.println(foo1.electricity + foo1.fluid);
14          foo1.electricity = "I declare ";
15          foo1.fluid = "a thumb war";
16          System.out.println(foo2.electricity + foo2.fluid);
17      }
18  }
```

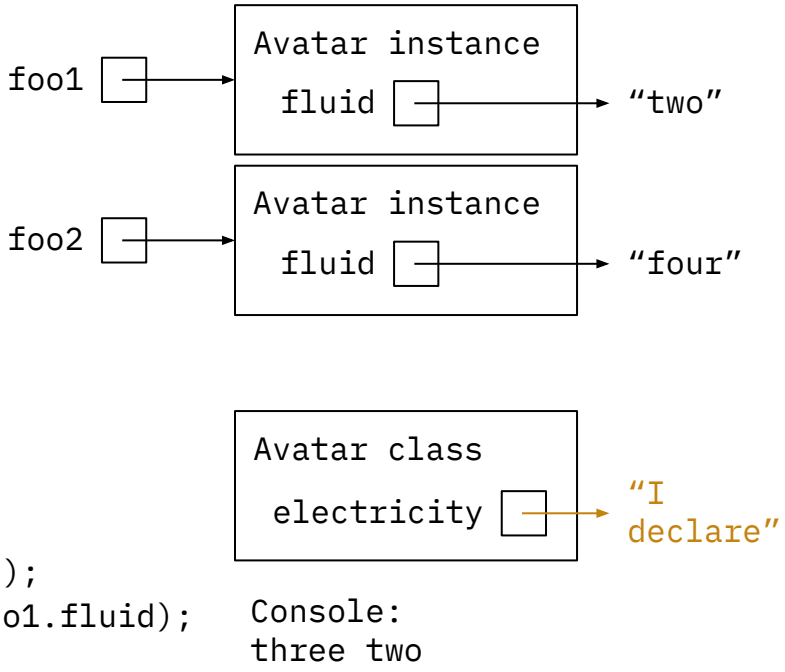


Console:
three two

What would be printed after executing the main method?

1A Objects Review

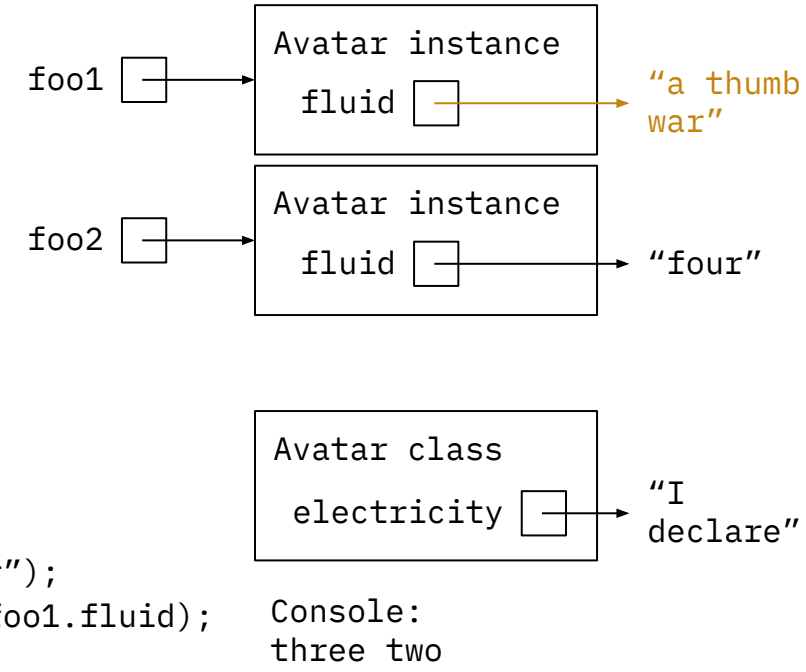
```
1   public class Avatar {
2       public static String electricity;
3       public String fluid;
4
5       public Avatar(String str1, String str2) {
6           Avatar.electricity = str1;
7           this.fluid = str2;
8       }
9
10      public static void main(String[] args) {
11          Avatar foo1 = new Avatar("one", "two");
12          Avatar foo2 = new Avatar("three", "four");
13          System.out.println(foo1.electricity + foo1.fluid);
14          foo1.electricity = "I declare ";
15          foo1.fluid = "a thumb war";
16          System.out.println(foo2.electricity + foo2.fluid);
17      }
18  }
```



What would be printed after executing the main method?

1A Objects Review

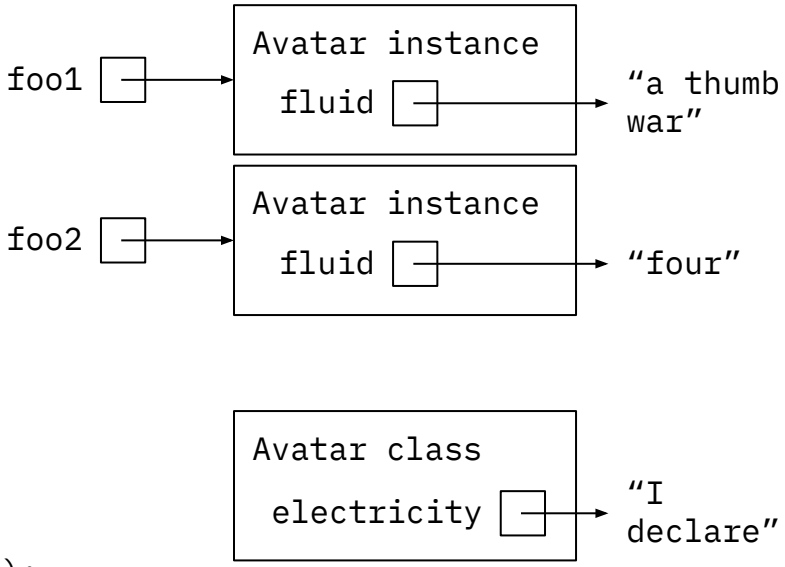
```
1   public class Avatar {
2       public static String electricity;
3       public String fluid;
4
5       public Avatar(String str1, String str2) {
6           Avatar.electricity = str1;
7           this.fluid = str2;
8       }
9
10      public static void main(String[] args) {
11          Avatar foo1 = new Avatar("one", "two");
12          Avatar foo2 = new Avatar("three", "four");
13          System.out.println(foo1.electricity + foo1.fluid);
14          foo1.electricity = "I declare ";
15          foo1.fluid = "a thumb war";
16          System.out.println(foo2.electricity + foo2.fluid);
17      }
18  }
```



What would be printed after executing the main method?

1A Objects Review

```
1   public class Avatar {
2       public static String electricity;
3       public String fluid;
4
5       public Avatar(String str1, String str2) {
6           Avatar.electricity = str1;
7           this.fluid = str2;
8       }
9
10      public static void main(String[] args) {
11          Avatar foo1 = new Avatar("one", "two");
12          Avatar foo2 = new Avatar("three", "four");
13          System.out.println(foo1.electricity + foo1.fluid);
14          foo1.electricity = "I declare ";
15          foo1.fluid = "a thumb war";
16          System.out.println(foo2.electricity + foo2.fluid);
17      }
18  }
```



Console:
three two
I declare four

What would be printed after executing the main method?

1B Objects Review

```
1  public class Avatar {
2      public ~~static~~ String electricity;
3      public static String fluid;
4
5      public Avatar(String str1, String str2) {
6          Avatar.electricity = str1;
7          this.fluid = str2;
8      }
9
10     public static void main(String[] args) {
11         Avatar foo1 = new Avatar("one", "two");
12         Avatar foo2 = new Avatar("three", "four");
13         System.out.println(foo1.electricity + foo1.fluid);
14         foo1.electricity = "I declare ";
15         foo1.fluid = "a thumb war";
16         System.out.println(foo2.electricity + foo2.fluid);
17     }
18 }
```

Would this code compile if we changed lines 2 and 3?

1B Objects Review

```
1  public class Avatar {
2      public ~~static~~ String electricity;
3      public static String fluid;
4
5      public Avatar(String str1, String str2) {
6          Avatar.electricity = str1; // Errors since it is now an instance variable
7          this.fluid = str2; // This is still fine!
8      }
9
10     public static void main(String[] args) {
11         Avatar foo1 = new Avatar("one", "two");
12         Avatar foo2 = new Avatar("three", "four");
13         System.out.println(foo1.electricity + foo1.fluid);
14         foo1.electricity = "I declare ";
15         foo1.fluid = "a thumb war";
16         System.out.println(foo2.electricity + foo2.fluid);
17     }
18 }
```

Would this code compile if we changed lines 2 and 3?

1C Objects Review

```
1   public class Avatar {
2       public static String electricity;
3       public String fluid;
4
5       ...
6
7       public static String getFluid() {
8           return fluid;
9       }
10  }
```

Would this code compile if we added this getFluid() function?

1C Objects Review

```
1   public class Avatar {
2       public static String electricity;
3       public String fluid;
4
5       ...
6
7       public static String getFluid() { // Compile-time error
8           return fluid; // Can't access fluid from a static function
9       }
10  }
```

Would this code compile if we added this getFluid() function?

2 Static Shock *Extra*

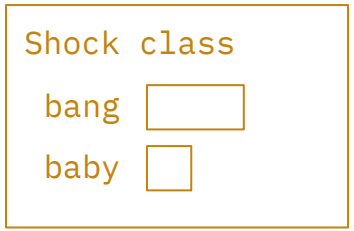
```
1   public class Shock {
2       public static int bang;
3       public static Shock baby;
4       public Shock() { this.bang = 100; }
5       public Shock(int num) {
6           this.bang = num;
7           baby = starter();
8           this.bang += num;
9       }
10      public static Shock starter() {
11          Shock gear = new Shock();
12          return gear;
13      }
14      public static void shrink(Shock statik) { statik.bang -= 1; }
15      public static void main(String[] args) {
16          Shock gear = new Shock(200);
17          System.out.println(gear.bang);
18          shrink(gear);
19          shrink(starter());
20          System.out.println(gear.bang);
21      }
22  }
```

What will the main method print?

2 Static Shock *Extra*

```
1   public class Shock {
2       public static int bang;
3       public static Shock baby;
4       public Shock() { this.bang = 100; }
5       public Shock(int num) {
6           this.bang = num;
7           baby = starter();
8           this.bang += num;
9       }
10      public static Shock starter() {
11          Shock gear = new Shock();
12          return gear;
13      }
14      public static void shrink(Shock statik) { statik.bang -= 1; }
15      public static void main(String[] args) {
16          Shock gear = new Shock(200);
17          System.out.println(gear.bang);
18          shrink(gear);
19          shrink(starter());
20          System.out.println(gear.bang);
21      }
22  }
```

gear 

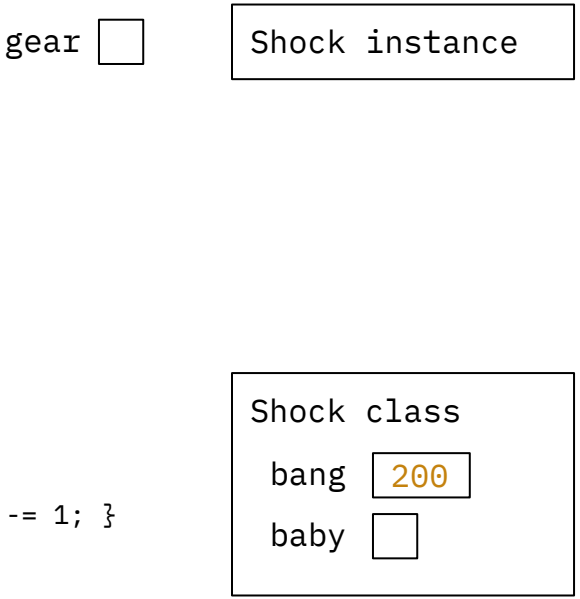


Console:

What will the main method print?

2 Static Shock *Extra*

```
1   public class Shock {
2       public static int bang;
3       public static Shock baby;
4       public Shock() { this.bang = 100; }
5       public Shock(int num) {
6           this.bang = num;
7           baby = starter();
8           this.bang += num;
9       }
10      public static Shock starter() {
11          Shock gear = new Shock();
12          return gear;
13      }
14      public static void shrink(Shock statik) { statik.bang -= 1; }
15      public static void main(String[] args) {
16          Shock gear = new Shock(200);
17          System.out.println(gear.bang);
18          shrink(gear);
19          shrink(starter());
20          System.out.println(gear.bang);
21      }
22  }
```

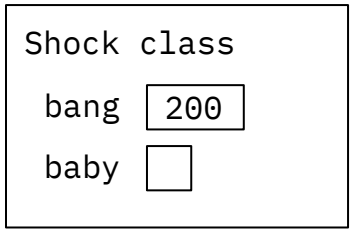


What will the main method print?

2 Static Shock *Extra*

```
1   public class Shock {
2       public static int bang;
3       public static Shock baby;
4       public Shock() { this.bang = 100; }
5       public Shock(int num) {
6           this.bang = num;
7           baby = starter();
8           this.bang += num;
9       }
10      public static Shock starter() {
11          Shock gear = new Shock();
12          return gear;
13      }
14      public static void shrink(Shock statik) { statik.bang -= 1; }
15      public static void main(String[] args) {
16          Shock gear = new Shock(200);
17          System.out.println(gear.bang);
18          shrink(gear);
19          shrink(starter());
20          System.out.println(gear.bang);
21      }
22  }
```

gear 

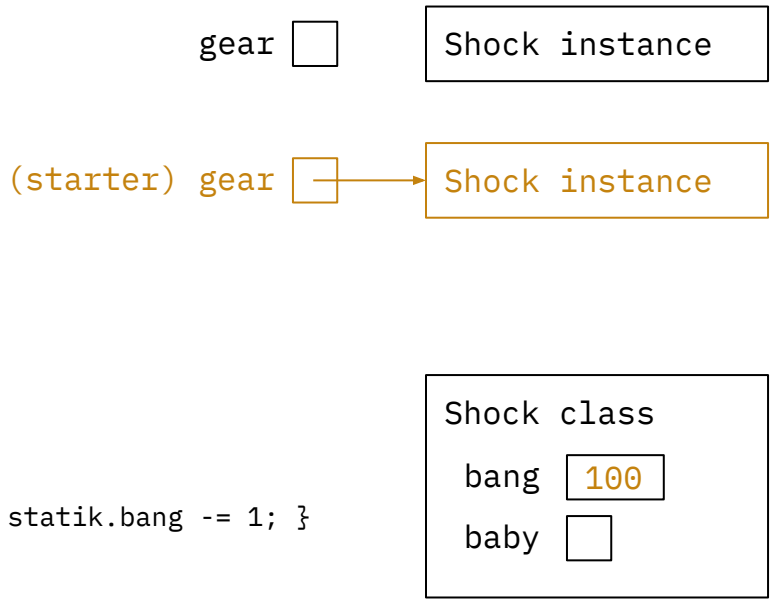


Console:

What will the main method print?

2 Static Shock *Extra*

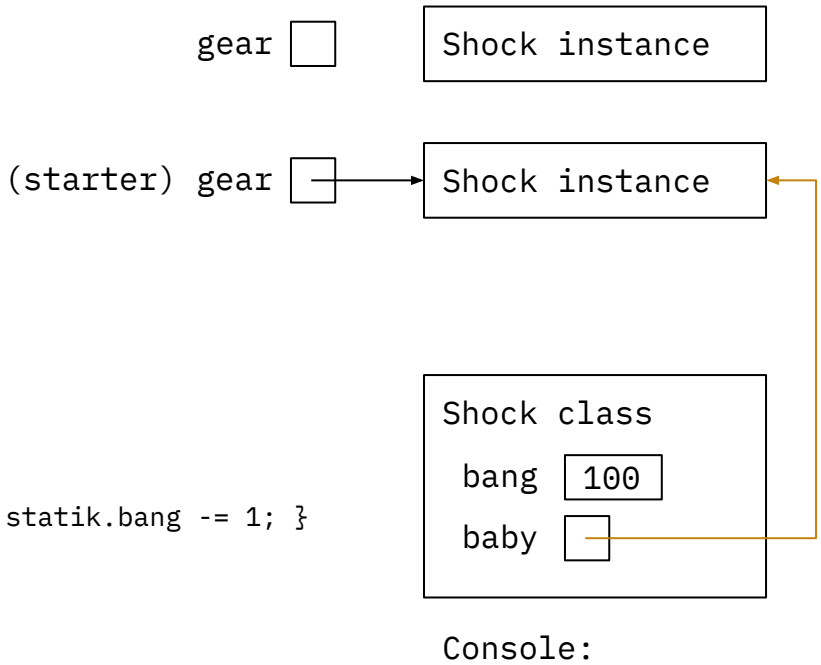
```
1   public class Shock {
2       public static int bang;
3       public static Shock baby;
4       public Shock() { this.bang = 100; }
5       public Shock(int num) {
6           this.bang = num;
7           baby = starter();
8           this.bang += num;
9       }
10      public static Shock starter() {
11          Shock gear = new Shock();
12          return gear;
13      }
14      public static void shrink(Shock statik) { statik.bang -= 1; }
15      public static void main(String[] args) {
16          Shock gear = new Shock(200);
17          System.out.println(gear.bang);
18          shrink(gear);
19          shrink(starter());
20          System.out.println(gear.bang);
21      }
22  }
```



What will the main method print?

2 Static Shock *Extra*

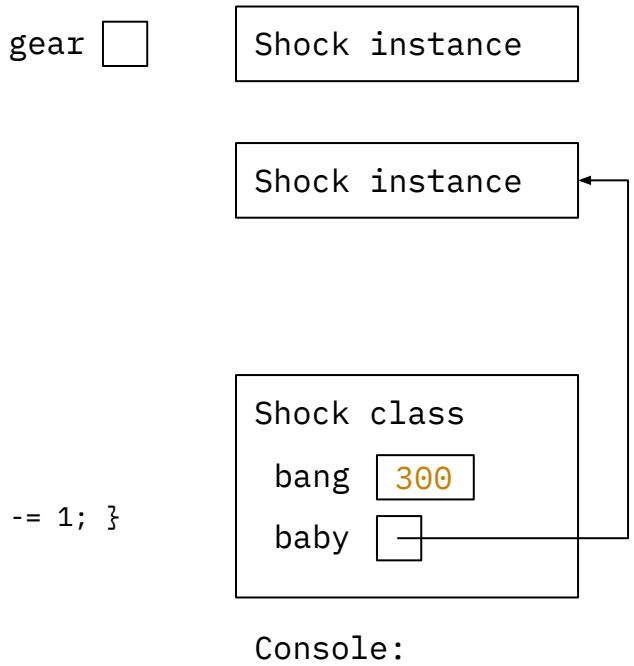
```
1   public class Shock {
2       public static int bang;
3       public static Shock baby;
4       public Shock() { this.bang = 100; }
5       public Shock(int num) {
6           this.bang = num;
7           baby = starter();
8           this.bang += num;
9       }
10      public static Shock starter() {
11          Shock gear = new Shock();
12          return gear;
13      }
14      public static void shrink(Shock statik) { statik.bang -= 1; }
15      public static void main(String[] args) {
16          Shock gear = new Shock(200);
17          System.out.println(gear.bang);
18          shrink(gear);
19          shrink(starter());
20          System.out.println(gear.bang);
21      }
22  }
```



What will the main method print?

2 Static Shock *Extra*

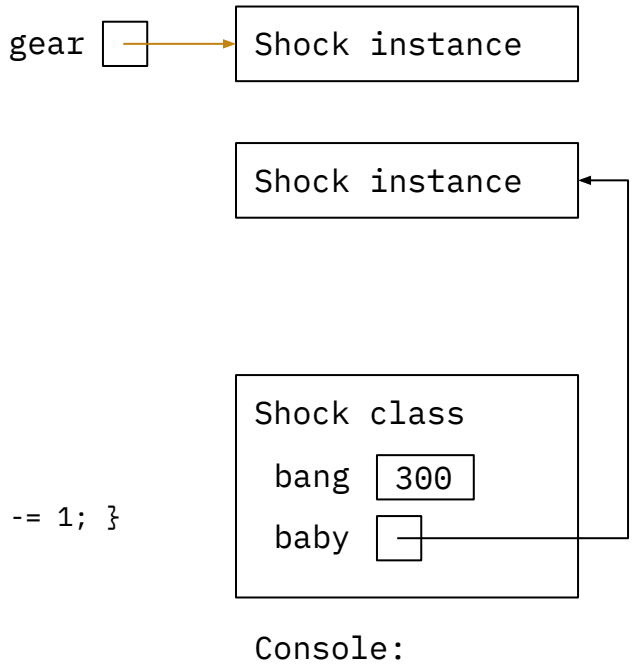
```
1   public class Shock {
2       public static int bang;
3       public static Shock baby;
4       public Shock() { this.bang = 100; }
5       public Shock(int num) {
6           this.bang = num;
7           baby = starter();
8           this.bang += num;
9       }
10      public static Shock starter() {
11          Shock gear = new Shock();
12          return gear;
13      }
14      public static void shrink(Shock statik) { statik.bang -= 1; }
15      public static void main(String[] args) {
16          Shock gear = new Shock(200);
17          System.out.println(gear.bang);
18          shrink(gear);
19          shrink(starter());
20          System.out.println(gear.bang);
21      }
22  }
```



What will the main method print?

2 Static Shock *Extra*

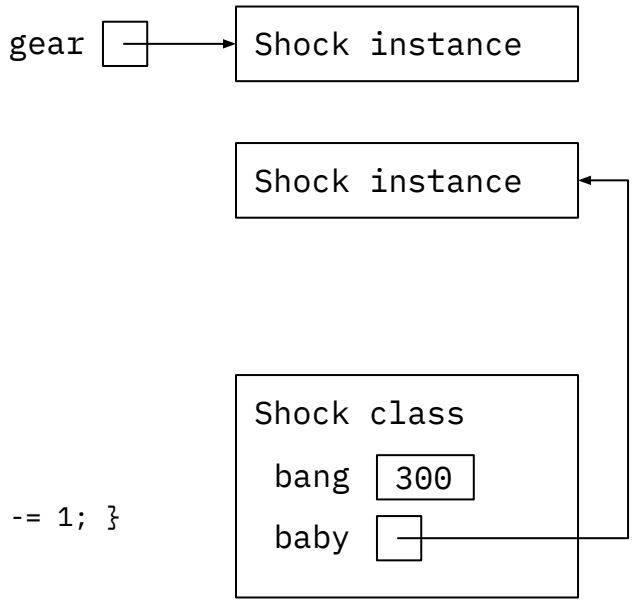
```
1   public class Shock {
2       public static int bang;
3       public static Shock baby;
4       public Shock() { this.bang = 100; }
5       public Shock(int num) {
6           this.bang = num;
7           baby = starter();
8           this.bang += num;
9       }
10      public static Shock starter() {
11          Shock gear = new Shock();
12          return gear;
13      }
14      public static void shrink(Shock statik) { statik.bang -= 1; }
15      public static void main(String[] args) {
16          Shock gear = new Shock(200);
17          System.out.println(gear.bang);
18          shrink(gear);
19          shrink(starter());
20          System.out.println(gear.bang);
21      }
22  }
```



What will the main method print?

2 Static Shock *Extra*

```
1   public class Shock {
2       public static int bang;
3       public static Shock baby;
4       public Shock() { this.bang = 100; }
5       public Shock(int num) {
6           this.bang = num;
7           baby = starter();
8           this.bang += num;
9       }
10      public static Shock starter() {
11          Shock gear = new Shock();
12          return gear;
13      }
14      public static void shrink(Shock statik) { statik.bang -= 1; }
15      public static void main(String[] args) {
16          Shock gear = new Shock(200);
17          System.out.println(gear.bang);
18          shrink(gear);
19          shrink(starter());
20          System.out.println(gear.bang);
21      }
22  }
```

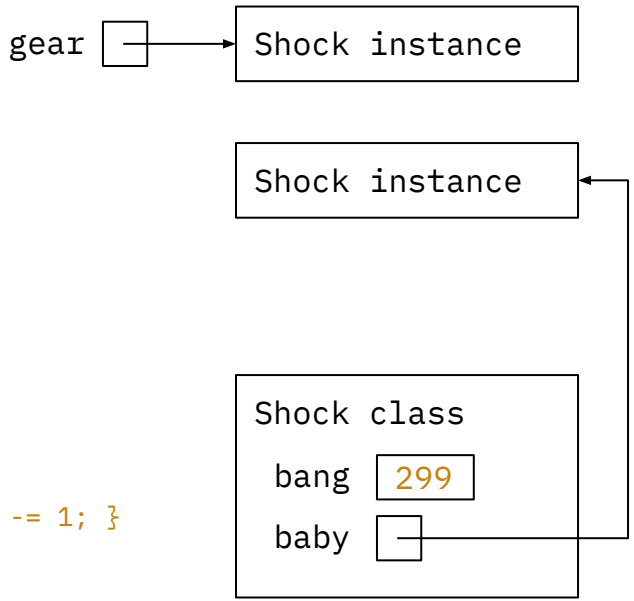


Console:
300

What will the main method print?

2 Static Shock *Extra*

```
1   public class Shock {
2       public static int bang;
3       public static Shock baby;
4       public Shock() { this.bang = 100; }
5       public Shock(int num) {
6           this.bang = num;
7           baby = starter();
8           this.bang += num;
9       }
10      public static Shock starter() {
11          Shock gear = new Shock();
12          return gear;
13      }
14      public static void shrink(Shock statik) { statik.bang -= 1; }
15      public static void main(String[] args) {
16          Shock gear = new Shock(200);
17          System.out.println(gear.bang);
18          shrink(gear);
19          shrink(starter());
20          System.out.println(gear.bang);
21      }
22  }
```

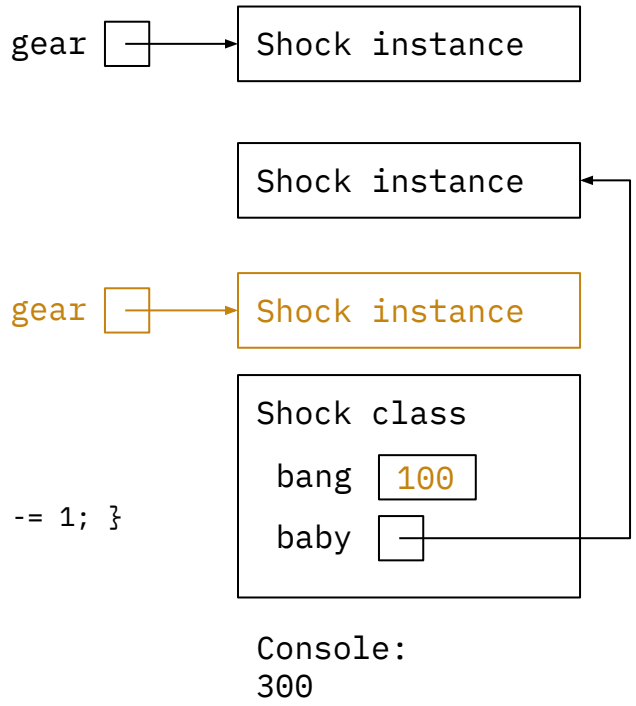


Console:
300

What will the main method print?

2 Static Shock *Extra*

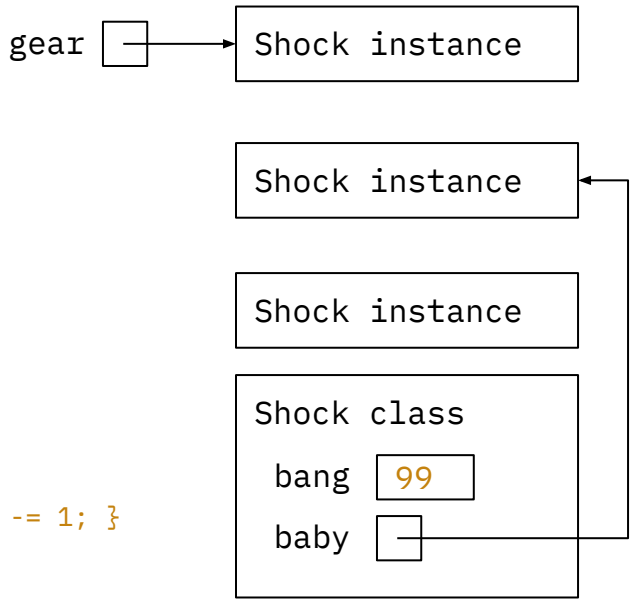
```
1   public class Shock {
2       public static int bang;
3       public static Shock baby;
4       public Shock() { this.bang = 100; }
5       public Shock(int num) {
6           this.bang = num;
7           baby = starter();
8           this.bang += num;
9       }
10      public static Shock starter() {
11          Shock gear = new Shock();
12          return gear;
13      }
14      public static void shrink(Shock statik) { statik.bang -= 1; }
15      public static void main(String[] args) {
16          Shock gear = new Shock(200);
17          System.out.println(gear.bang);
18          shrink(gear);
19          shrink(starter());
20          System.out.println(gear.bang);
21      }
22  }
```



What will the main method print?

2 Static Shock *Extra*

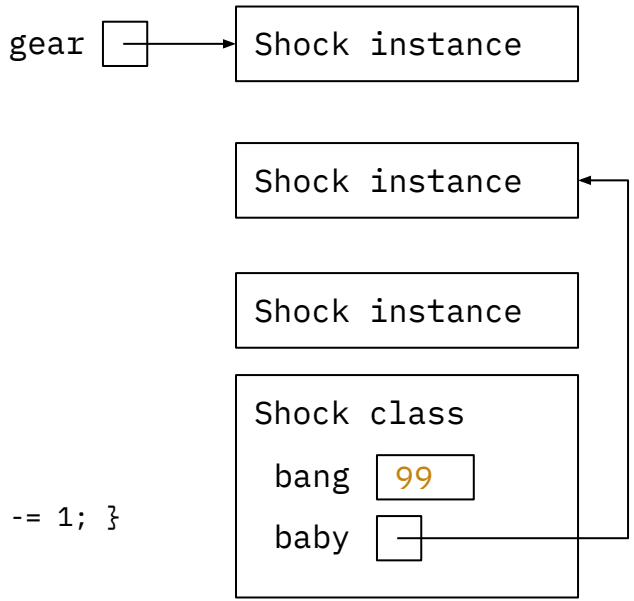
```
1   public class Shock {
2       public static int bang;
3       public static Shock baby;
4       public Shock() { this.bang = 100; }
5       public Shock(int num) {
6           this.bang = num;
7           baby = starter();
8           this.bang += num;
9       }
10      public static Shock starter() {
11          Shock gear = new Shock();
12          return gear;
13      }
14      public static void shrink(Shock statik) { statik.bang -= 1; }
15      public static void main(String[] args) {
16          Shock gear = new Shock(200);
17          System.out.println(gear.bang);
18          shrink(gear);
19          shrink(starter());
20          System.out.println(gear.bang);
21      }
22  }
```



What will the main method print?

2 Static Shock *Extra*

```
1   public class Shock {
2       public static int bang;
3       public static Shock baby;
4       public Shock() { this.bang = 100; }
5       public Shock(int num) {
6           this.bang = num;
7           baby = starter();
8           this.bang += num;
9       }
10      public static Shock starter() {
11          Shock gear = new Shock();
12          return gear;
13      }
14      public static void shrink(Shock statik) { statik.bang -= 1; }
15      public static void main(String[] args) {
16          Shock gear = new Shock(200);
17          System.out.println(gear.bang);
18          shrink(gear);
19          shrink(starter());
20          System.out.println(gear.bang);
21      }
22  }
```



Console:
300
99

What will the main method print?

3A Reversing an Array

Implement reverse such that it destructively reverses the elements of A.

```
public static void reverse (int[] A) {




}
```

3A Reversing an Array

Implement reverse such that it destructively reverses the elements of A.

```
public static void reverse (int[] A) {




}
// First element needs to be the last element and vice versa
// Same pattern for the entire array
```

3A Reversing an Array

Implement reverse such that it destructively reverses the elements of A.

```
public static void reverse (int[] A) {
    for (int i = 0; i < A.length / 2; i++) {
        // Only need to loop through half of the array
        // Other half gets solved as we swap

    }
}
```

3A Reversing an Array

Implement reverse such that it destructively reverses the elements of A.

```
public static void reverse (int[] A) {
    for (int i = 0; i < A.length / 2; i++) {
        int temp = A[A.length - i - 1]; // Now swap!
        A[A.length - i - 1] = A[i];
        A[i] = temp;
    }
}
```

3B Reversing an Array *Extra*

Implement **reverseDiagonal** such that it destructively reverses the elements of **B** along the diagonal.

```
public static void reverseDiagonal (int[][] B, int diagonal) {



}
```

3B Reversing an Array *Extra*

Implement reverseDiagonal such that it destructively reverses the elements of B along the diagonal.

```
public static void reverseDiagonal (int[][] B, int diagonal) {




}
// Same idea as last problem except across the diagonal
// The nth diagonal has n+1 terms in it (check yourself!)
```

3B Reversing an Array *Extra*

Implement reverseDiagonal such that it destructively reverses the elements of B along the diagonal.

```
public static void reverseDiagonal (int[][] B, int diagonal) {
    for (int i = 0; i <= diagonal / 2; i++) { // Iterate through the diagonal



    }
}
```

3B Reversing an Array *Extra*

Implement reverseDiagonal such that it destructively reverses the elements of B along the diagonal.

```
public static void reverseDiagonal (int[][] B, int diagonal) {
    for (int i = 0; i <= diagonal / 2; i++) {
        int temp = B[diagonal - i][i]; // Diagonal so we need both coordinates


    }
}
```

3B Reversing an Array *Extra*

Implement reverseDiagonal such that it destructively reverses the elements of B along the diagonal.

```
public static void reverseDiagonal (int[][] B, int diagonal) {
    for (int i = 0; i <= diagonal / 2; i++) {
        int temp = B[diagonal - i][i];
        B[diagonal - i][i] = B[i][diagonal - i];
        B[i][diagonal - i] = temp; // Finish swapping
    }
}
```

4 Circular Buffer

Implement overflow such that it non-destructively flattens the circular buffer.

```
public static int[] overflow (int[] A, int i, int k) {




}
```

4 Circular Buffer

Implement overflow such that it non-destructively flattens the circular buffer.

```
public static int[] overflow (int[] A, int i, int k) {
    int[] B = new int[A.length + 1]; // Create new array that's one bigger




}
```

4 Circular Buffer

Implement overflow such that it non-destructively flattens the circular buffer.

```
public static int[] overflow (int[] A, int i, int k) {
    int[] B = new int[A.length + 1];
    System.arraycopy(A, i, B, 0, A.length - i); // Copying “beginning” of circular buffer



}
```

4 Circular Buffer

Implement overflow such that it non-destructively flattens the circular buffer.

```
public static int[] overflow (int[] A, int i, int k) {
    int[] B = new int[A.length + 1];
    System.arraycopy(A, i, B, 0, A.length - i);
    System.arraycopy(A, 0, B, A.length - i, i); // Copying “end” of circular buffer


}
```

4 Circular Buffer

Implement overflow such that it non-destructively flattens the circular buffer.

```
public static int[] overflow (int[] A, int i, int k) {
    int[] B = new int[A.length + 1];
    System.arraycopy(A, i, B, 0, A.length - i);
    System.arraycopy(A, 0, B, A.length - i, i);
    B[A.length] = k; // Insert new item

}
```

4 Circular Buffer

Implement overflow such that it non-destructively flattens the circular buffer.

```
public static int[] overflow (int[] A, int i, int k) {
    int[] B = new int[A.length + 1];
    System.arraycopy(A, i, B, 0, A.length - i);
    System.arraycopy(A, 0, B, A.length - i, i);
    B[A.length] = k;
    return B; // return our new array
}
```

5 Transposing a 2D Array *Extra*

Implement transpose such that it destructively transposes two dimensional array input.

```
public static void transpose (int[][] A) {




}
```

5 Transposing a 2D Array *Extra*

Implement transpose such that it destructively transposes two dimensional array input.

```
public static void transpose (int[][] A) {
    for (int i = 0; i < A.length; i++) { // Iterate through everything length-wise




    }
}
```

5 Transposing a 2D Array *Extra*

Implement transpose such that it destructively transposes two dimensional array input.

```
public static void transpose (int[][] A) {
    for (int i = 0; i < A.length; i++) {
        for (int j = i; j < A[i].length; j++) { // Iterate through everything height-wise



        }
    }
}
```

5 Transposing a 2D Array *Extra*

Implement transpose such that it destructively transposes two dimensional array input.

```
public static void transpose (int[][] A) {
    for (int i = 0; i < A.length; i++) {
        for (int j = i; j < A[i].length; j++) {
            int temp = A[j][i]; // Swap diagonally
            A[j][i] = A[i][j];
            A[i][j] = temp;
        }
    }
}
```

5 Transposing a 2D Array *Extra*

Implement transpose such that it destructively transposes two dimensional array input.

```
public static void transpose (int[][] A) {
    for (int i = 0; i < A.length; i++) {
        for (int j = i; j < A[i].length; j++) {
            int temp = A[j][i];
            A[j][i] = A[i][j];
            A[i][j] = temp;
        }
    }
}
```