

# Hidden Fruits

[Here is a video walkthrough of the solutions.](#)

Suppose we have the `Fruit` and `Persimmon` and classes defined below.

```
1  class Fruit {
2      String flavor = "generic";
3      static char start = 'f';
4
5      static int eat(Fruit fruit) {
6          return 1;
7      }
8
9      char hats() {
10         return this.start;
11     }
12 }
13
14 class Persimmon extends Fruit {
15     String flavor = "superb";
16     static char start = 'p';
17
18     static int eat(Fruit fruit) {
19         return 2;
20     }
21
22     int eat(Persimmon persimmon) {
23         return 3;
24     }
25 }
```

For each line below, write what, if anything, is printed after its execution. Write CE if there is a compiler error and RE if there is a runtime error. If a line errors, continue executing the rest of the lines.

```
1  Fruit shreyas = new Fruit();
2  Fruit aram = new Persimmon();
3  Persimmon eric = new Persimmon();
4
5  System.out.println(eric.flavor);
6  System.out.println(aram.flavor);
7
8  System.out.println(eric.eat(shreyas));
9  System.out.println(eric.eat(eric));
10 System.out.println(aram.eat(eric));
11
12 System.out.println(aram.hats());
13 System.out.println(eric.hats());
```

**Solution:**

```
Fruit shreyas = new Fruit();
Fruit aram = new Persimmon();
Persimmon eric = new Persimmon();

System.out.println(eric.flavor); // superb
System.out.println(aram.flavor); // generic

System.out.println(eric.eat(shreyas)); // 2
System.out.println(eric.eat(eric)); // 3
System.out.println(aram.eat(eric)); // 1

System.out.println(aram.hats()); // f
System.out.println(eric.hats()); // f
```