Exercise 1

The point of this exercise is to practice reading code and to make sure that you understand the flow of execution through a program with multiple methods.

```
public static void zoop() {
    baffle();
    System.out.print("You wugga ");
    baffle();
}

public static void main(String[] args) {
    System.out.print("No, I ");
    zoop();
    System.out.print("I ");
    baffle();
}

public static void baffle() {
    System.out.print("wug");
    ping();
}

public static void ping() {
    System.out.println(".");
}
```

1. What is the output of the following program? Be precise about where there are spaces and where there are newlines.
   Hint: Start by describing in words what `ping` and `baffle` do when they are invoked.

   zoop() calls baffle(), prints out “You wugga “ without a newline character, and then calls baffle() again.

   baffle() prints out “wug” without a newline and calls ping().

ping() prints “.” with a newline.

zoop():
wug.
You wuggawug.
(newline)

baffle():
wug.
(newline)

ping():
.
(newline)

main():
No I wug.
You wuggawug.
I wug.

2. Draw a stack diagram that shows the state of the program the first time `ping` is invoked.

main()
zoop()
baffle()
ping()

3. What happens if you invoke `baffle();` at the end of the `ping` method? (We will see why in the next chapter.)

Since baffle() calls ping() you will end up with recursion baffle()-> ping() -> baffle() -> ping() and since there is no base case to end the recursion you will have an infinite loop.