Functions

Important notes for writing better functions:

1. First rule of functions is that they should be small. Second rule is that they should be smaller than that.

2.blocks within if , else and while should be one line long. Probably the line should be a function call.

3.functions should do one thing

4.the functions should be followed by those at next level of abstraction, so that it can be read by descending one level of abstraction as we read down the list of functions (this is step-down rule)

5.Listing 3-4 has problems such as, the function is too large, it does not do one thing, it violates Single Responsibility Principle(SRP) because there is more than one reason for it to change, it violates Open Closed Principle(OCP) because it must change whenever new types are added and also there will be several other functions that follow same structure.

6.the solution to above problem in listing code is to bury the switch statement in ABSTRACT FACTORY.

7.output arguments are harder to understand than input arguments. We use the idea of information going out of function through return value.

8.flag arguments are ugly. Because it clearly does more than one thing one if flag is true and other if flag is false.

9.if a function needs more than two or three arguments, it is likely that some of arguments ought to be wrapped into a class of their own.

10.assertExpectedEqualsActual(expected, actual) mitigate the problem of having to remember the order of arguments.

11.Anything that forces you to check the function signature is equivalent to double-take. It’s a cognitive break and should be avoided.

12.Returning error codes from command functions is subtle violation of command query separation

13.Try/Catch blocks are ugly. They confuse the structure of the code and mix error processing with normal processing. So it is better to extract the bodies of the try catch blocks out into functions of their own.