Refactoring

Code Smells

COMP3607
Object Oriented Programming II

Code Smells

“A surface indication that usually corresponds to a deeper problem in the software system” - Martin Fowler

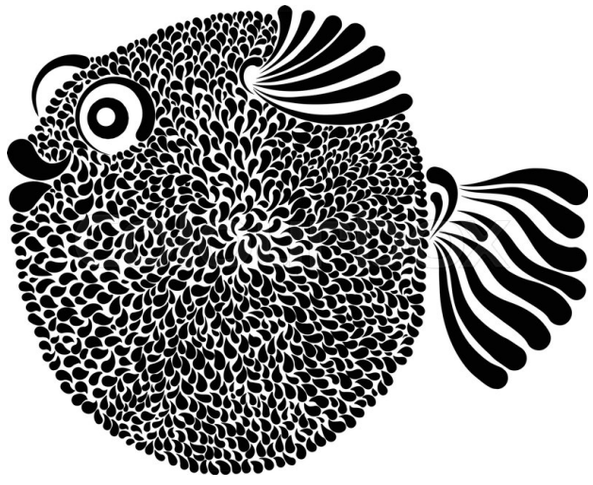
Bloaters

Bloaters are code, methods and classes that have increased to such proportions that they are hard to work with.

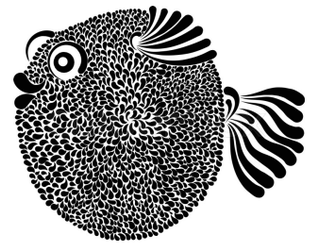
Usually these smells do not crop up right away, rather they accumulate over time as the program evolves.

For example:

- Long Method
- Large Class
- Primitive Obsession
- Long Parameter List
- Data Clumps.



Long Method



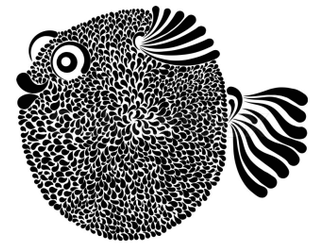
The majority of a programmer's time is spent reading code rather than writing code.

Apart from the difficulty of having to keep a lot of complex logic in mind whilst reading through a long method, it is usually a sign that the method has too many responsibilities.

Long methods make code hard to maintain and debug.

If it is not possible to view the whole method on your smartphone screen, consider breaking it up into several smaller methods, each doing one precise thing.

Large Class

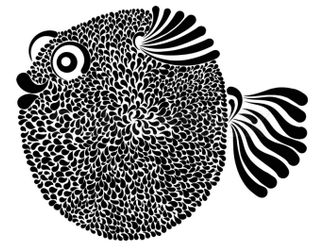


When a single class is doing too much, it often shows up as too many variables, instances and methods.

Classes usually start small but over time they get bloated as the program grows.

Programmers usually find it mentally less taxing to place a new feature in an existing class rather than create a new class for the feature.

Data Clumps



Where multiple method calls take the same set of parameters, it may be a sign that those parameters are related.

To keep the group of parameters together, it can be useful to combine them together in a class.

This can help aid organisation of code.

References

- Martin Fowler. Refactoring: Improving the Design of Existing Code
- http://wiki.c2.com/?CodeSmell