# The next 700 “bad” languages imagined.

Camille Akmut

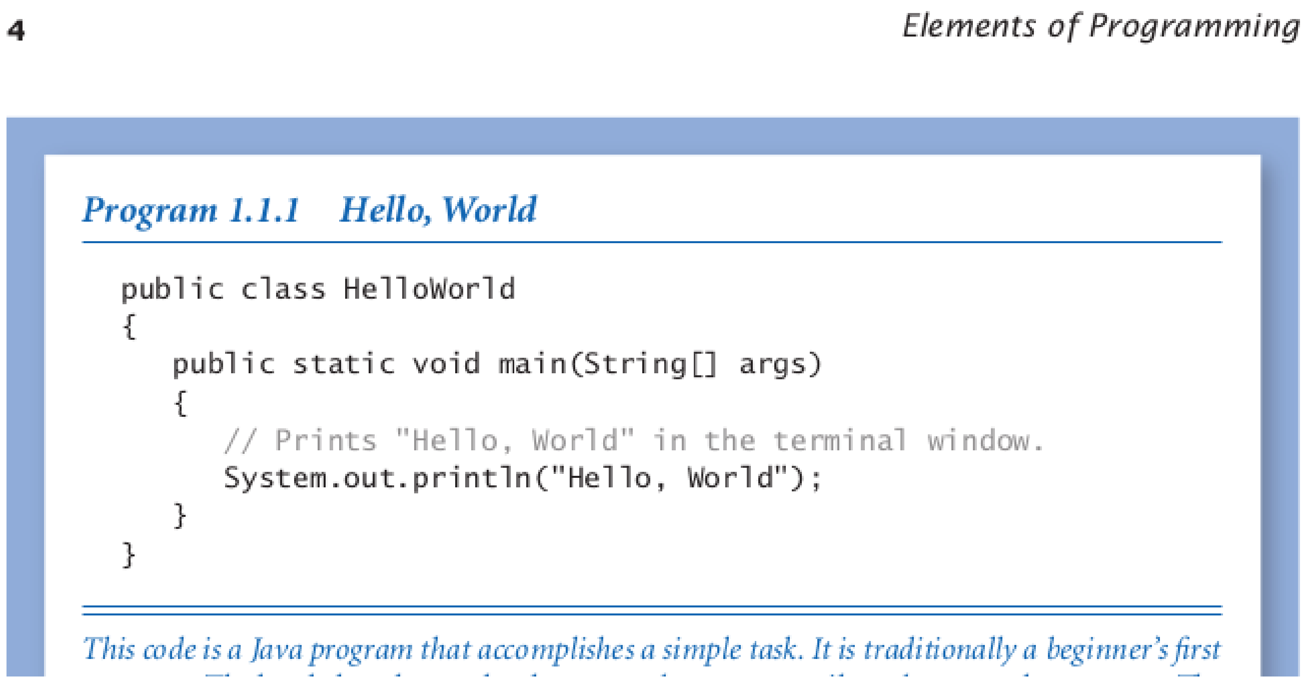
**Abstract**

More political and technical notes on programming languages. So future generations won’t have to suffer through another Java or JavaScript, or worse even.

# TAW : The awful ways of bad languages.

18-Aug-19

**I. A simple task is complicated — or the Java "Hello, World!" program…**



4 *Elements of Programming*

*Program 1.1.1 Hello, World*

```
public class HelloWorld
{
   public static void main(String[] args)
   {
      // Prints "Hello, World" in the terminal window.
      System.out.println("Hello, World");
   }
}
```

*This code is a Java program that accomplishes a simple task. It is traditionally a beginner's first*

Imagine the creators of Java, having reached the end of their painful journeys, coming up with such a "Hello, World!" program and nonetheless still agreeing in unison : "*we've done a good job*", "*we can go home now*", "*this is the language that should be defining for an entire decade*". And, now imagine countless computer science professors and administrators agreeing to the same… Such events could only be the product of a badly upside-down culture.

And, now imagine too the embarrassment of instructors having to earnestly explain to wide-eyed freshmen what `public static void main(String[] args)` means during a first semester's first class, and why this would be needed at such an early stage… But, you don't to have imagine if from this century. And, later the obsession of this language's designer with "public" and "private" keywords (for security), and their pathological reliance on classes for anything and everything under the sun.

And, imagine too, if p. 4 of this canonical textbook — "*a simple task*" — is like this, what page 404 looks like. (Another error, in a long list.)

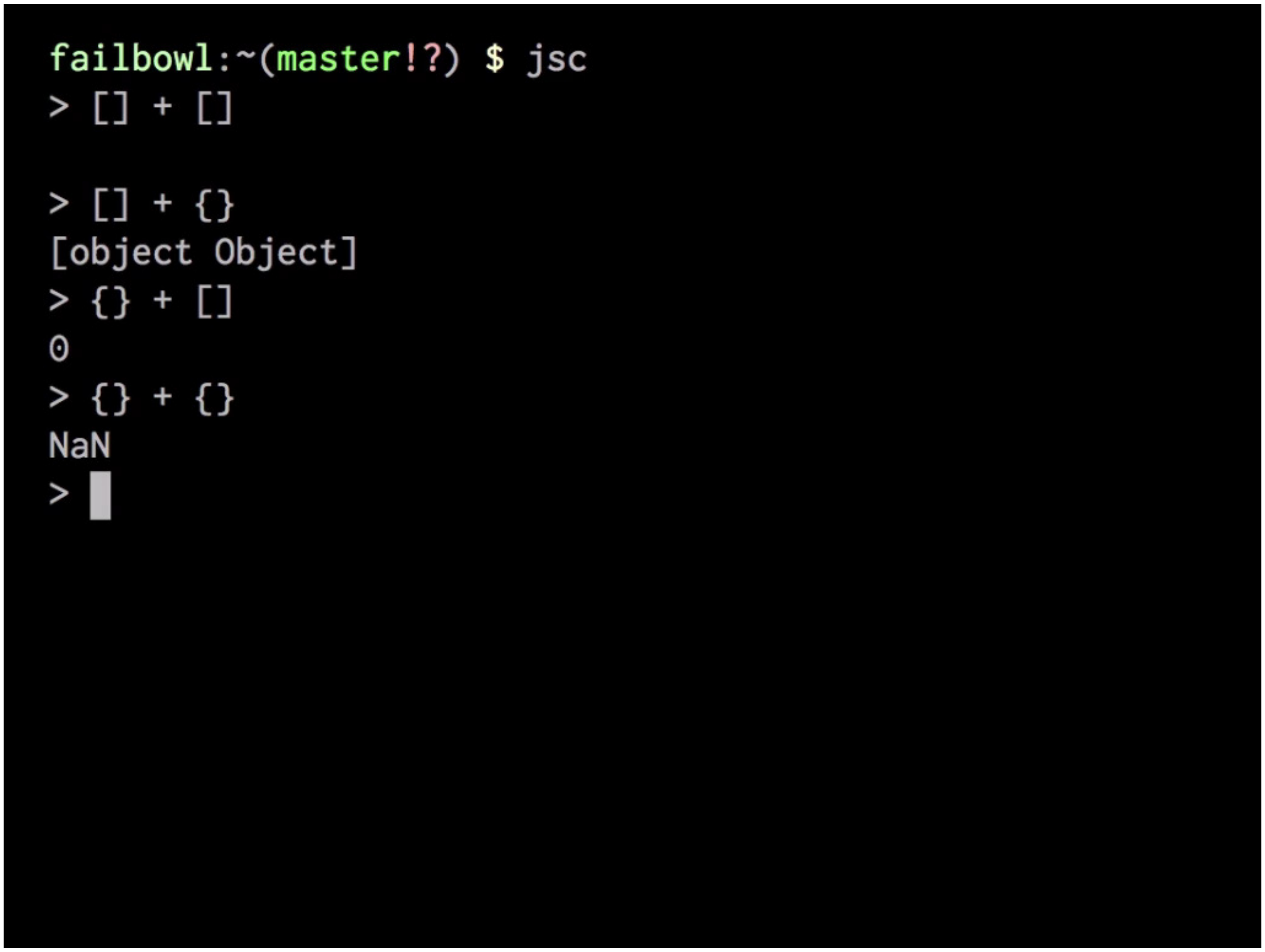
**II. A third equals sign — or the abysmal "===".**

If during the design of a language a third equals sign somehow emerges as a necessity, then without a doubt something happened at some point — somewhere, somehow — that was tremendously, tremendously wrong (most likely to do with the underlying type system, its weakness.).

Two operations are fundamental : assignment and equality, represented reasonably by e.g. “=” and “==” (perhaps “:=” and “=”).

A triple equals sign (“===”) is the sure sign by which any “bad” language identifies itself; product of the two abysmal languages of this epoch.

**III. Languages with “*good parts*“ / Does not survive a 4 min. video.**



```
failbowl:~(master!?) $ jsc
> [] + []

> [] + {}
[object Object]
> {} + []
0
> {} + {}
NaN
> 
```

Here, so little makes sense, and the general lack of coherence is so great that one is left to wonder what went through the minds of its creators. These languages cannot pretend to their titles (“non-languages”, rather); though unfortunately communication in them is sometimes widespread.

In the first case, either a type error, or an empty list (result of concatenating two empty lists e.g. `[] ++ []` in Haskell) might have been expected results.

But, if nothing is predictable, how does anyone expect efficient communication to be possible in them, let alone reliable software to be written in them? (despite the insistence of self-identified “coders” in these languages on “engineer” titles, unlike anything seen in other languages and programmers.)

An ‘expertise’ in them stems only from knowing a long list of what to avoid — another tell-all sign of any “bad” language, also known as languages with “*good parts*“…

**IV. Languages with distinctive corporate allures and identity.**

Among “bad” languages are consistently those that rely on aggressive marketing and corporate power to establish themselves (rather than by relying on their intrinsic qualities, and communities).

e.g. Java, Go …

As already pointed out by Edsger Dijkstra, who did not lack a sociological eye contrary to too many of his half-blind peers, “*Java (…) is a mess (and needed an extensive advertizing campaign and aggressive salesmanship for its commercial acceptance).*“ He would have said the same about the new Java of this generation — Go, as we have argued — and the next ones after.

**V. “Greedy” licensing — Miranda and David Turner’s misfortunes.**

In the end, no matter the merits of any language, if licensing prohibits the emergence of communities around it, that language will know a rapid death or decline; its ideas will be recuperated by other languages, language designers.

## References

"*Towards the end he became convinced that computing had perhaps been a bad idea, giving support to profit-taking corporate interests and a surveillance state, and that perhaps he'd wasted his energies in promoting it.*" (on Peter Landin)

—. 2019. "What is Computer Science?".

Bernhardt, Gary. 2012. "Wat". *CodeMash.*

Bornat, Richard. 2009. "Peter Landin obituary". *The Guardian*, 22/09.

Dijkstra, Edsger. 2001. "To the members of the Budget Council". https://www.cs.utexas.edu/users/EWD/transcriptions/OtherDocs/Haskell.html

Landin, Peter. 1966. "The next 700 programming languages". *Communications of the ACM* 9(3) : 157-166.

Sedgewick, Robert and Wayne, Kevin. 2017. *Computer Science. An interdisciplinary approach.* Addison-Wesley.

## Illustrations

1 : Sedgewick/Wayne 2017. 2 : Bernhardt 2012.