The unanimous Declaration of Programmers of A-Tech from Object-Oriented Design,

September 24, 2023.

When in the Course of intellectual events, it becomes necessary for one group of programmers to dissolve stylistic bands which have connected them with a paradigm, and to assume among the powers of the internet, which the laws of computing entitle them, a decent respect to the opinions of framework and language designers requires that they should declare the causes which impel them to the separation.

We hold these truths to cause the least technical debt, that variables are created equal, that they have certain use cases, that among these are Syntactical Elegance, Memory Safety, and the pursuit of Immutability; that to secure these uses, design patterns are implemented by language designers, from the requests of developers (though sometimes the Program Manager gets in the way). When these design patterns began to slow progress, it is the right of the Programmers to change or violate them, until progress can continue. Such has been the troubles of the programmers of this school, and why change must be done to the current programming strategies. The history of the present dominant programming styles is a history of the sudden frenzies programmers blindly follow in hopes their purposeless spaghetti code will compile, which lead to useless practices seen today. To prove this, let the following anti-patterns be revealed to the programming world.

OO has scared programmers from investigating more efficient paradigms, it has kept them in a bubble of comfort, perpetuated by Sun, Oracle, and Microsoft.

OO leads to painfully unnecessary syntactical constructions. The day "public static void main(String[] args)" becomes something convenient for programmers to type is the day all signs of human progress have left the face of the Earth.

OO, what the heck is even a "DoIFactoryDecoratorExecutorThread"?

OO pushes programmers to arbitrarily segregate their mutable state, without providing any solution to handling what makes mutable state difficult.

OO forcefully combines functions and data in ways that often have no purpose. How often do physical objects with behaviors manifest themselves in code?

Inheritance is simply a programming sin, endlessly proven to be inferior to composition. To believe otherwise is to only imagine the cats and dogs being inherited from the animal class.

Polymorphism is not exclusive to the domain of OO. In fact, other forms of polymorphism are more applicable to programming situations.

Abstraction has to be done meaningfully and with care, OO doesn't enable such forms of abstraction.

OO forces even the simplest units of data to become objects, not only is this unnecessary, but also prevents the better programming method. Having many functions work on few data structures, rather than many data structures on few functions.

Every time these problems have come up, we have tried to work around them, but endless work-arounds are what it means to have unpleasant code! We have tried modifying these programming practices to fit our goals, coming up with endless design strategies, and development frameworks, but time and time again these prove to be ineffective at addressing the true problem, OO programming itself.

We, therefore, the Programmers of Advanced Technologies Academy, Assembled, appeal to all programmers that we declare ourselves free and independent from Object-Oriented Programming and Design. We no longer follow the programming practices recommended by Collegeboard, Java Developers or any proponent of OOP. We have the right to make our behaviors separate from our data, and the right to choose the programming language that best aligns with our project's needs. And for the support of this Declaration, with a firm reliance on the Monad, we mutually pledge to each other our programming style, our projects and our sacred Honor.