What is the impact of programming on our world? Does everyone need to learn programming? How might individual lives and society as a whole be changed if we found a way to include programming and technology at all levels of education? It is acceptable in your essay to disagree and say that programming should not be for everybody. Make sure that whatever position you take, that you support your argument.

The impact of programming on our world is seen in nearly every aspect of our daily lives. Java code may be running the functions of your alarm clock and coffee maker, awakening you and brewing your morning coffee at a preset time. We use smartphones to check the weather, news, email and social media – all before getting out of bed. We have seen an explosion in technology all brought about by the skills of programmers who developed the software to drive these devices, and in turn drive device developers to increase the capabilities of their products.

Economically, we see the impact of programming in the buildup to keynote presentations by companies like Apple and Microsoft. These are big media events that have an impact on not only these companies’ stock prices, but also the entire sector. Right now, the media is buzzing about the arrival of Apple Watch. Debate about whether it will be a huge success or totally flop is the topic du jour for many technology journalists. We also see the economic impact of programming in the huge initial public offerings of companies like Facebook, Twitter and LinkedIn, as well as in the enduring high valuation placed on Google. Programming fuels ideas, and the conversion of an idea into a tool through programming is extremely valuable.

But should everyone learn programming? That is a more challenging question to consider. [Jeff Atwood](http://en.wikipedia.org/wiki/Jeff_Atwood), founder of Stack Overflow, believes that teaching programming to everyone is not appropriate. In an [NPR All Tech Considered article](http://www.npr.org/blogs/alltechconsidered/2014/01/25/266162832/computers-are-the-future-but-does-everyone-need-to-code), he expressed his disagreement by likening learning programming to becoming an auto mechanic. His point being that you don’t have to know the inner workings of the computer in order to use it as a tool in creating things.

Contrarily, we have Steve Jobs’ now infamous quote, “*I think everyone should learn how to program a computer, because it teaches you how to think. I view computer science as a liberal art, something everyone should learn to do.*" The Huffington Post article entitled, “[Why Teach Computer Science in High School](http://www.huffingtonpost.com/brian-heese/why-teach-computer-science_b_4350315.html)” described the benefits of learning programming to include attention to detail, applying logic, persistence, collaboration, and the ability to ask good questions. [Code.org’s](http://www.code.org) “[Hour of Code](http://hourofcode.com/us)” [video campaign](http://www.youtube.com/watch?v=nKIu9yen5nc) features celebrities along with tech icons like Mark Zuckerberg and Bill Gates encouraging people to pursue programming. The advocates for it are many and are very outspoken.

I think that Mark Zuckerberg’s comments in the Code.org video provide important insights to the benefits of teaching programming at all levels of education. He said, “*When I needed to learn something new I looked it up either in a book or on the internet and I added something to it* [the program].” Zuckerberg expanded on Steve Jobs’ concept of programming teaching you how to think. In essence, learning programming teaches you how to learn independently. The student learns how to seek answers and consume multiple sources of information in that effort. This is the reason that teaching programming at all levels of education is important.

What is unique about programming is that it allows for multiple paths to the same conclusion. You can write a program that functions properly and produces the same result in a variety of ways. In other disciplines, like mathematics, there are clearly right and wrong answers. You can solve countless problems, but you’ll always end up using the same processes for doing so. For instance, the order of operations will always be “[*Please Excuse My Dear Aunt Sally*](http://blogs.edweek.org/teachers/coach_gs_teaching_tips/2011/01/math_teachers_please_excuse_dear_aunt_sally--forever.html)”. Learning programming has elements of the inflexibility of mathematics, specifically syntax, but it is far more fluid. Two programmers can approach the same problem differently and still produce the same correct result.

Another important benefit is that programming changes the perspective on “failure” for the student. Initial programs are inherently buggy because students have to think of all of the possible ways that the user could respond to the program. Each time a user does something unexpected causing a bug, the programmer improves their ability to foresee potential pitfalls and prevent them in the code and builds upon the skills and knowledge they’ve gained.

In conclusion, the economic impact of programming has changed its perception from the domain of “nerds” to mainstream coolness. The use of the computer as a tool for creating things doesn’t require students to learn programming. However, learning to program offers many benefits that apply across other disciplines. Introducing programming at all levels of education serves the student by expanding their ability to foresee challenges and connect related ideas. Lastly, learning to program enables the student to learn independently, because there is no prescribed way to approach a programming problem. Programming encourages exploration by the student, and kindling the desire to learn through independent exploration is far more valuable than anything else that the student can be taught.

References:

Code.org video <http://www.youtube.com/watch?v=nKIu9yen5nc>

Why Teach Computer Science in High School <http://www.huffingtonpost.com/brian-heese/why-teach-computer-science_b_4350315.html>

Hour of Code <http://hourofcode.com/us>

Computers Are The Future, But Does Everyone Need To Code? <http://www.npr.org/blogs/alltechconsidered/2014/01/25/266162832/computers-are-the-future-but-does-everyone-need-to-code>

Please Excuse My Dear Aunt Sally (PEMDAS)-- forever! http://blogs.edweek.org/teachers/coach\_gs\_teaching\_tips/2011/01/math\_teachers\_please\_excuse\_dear\_aunt\_sally--forever.html