

# Has Programming become too easy?

BRUCE SPENCE - 2721301

Has programming become too easy?

## Contents

<b>1.0 - Position</b> .....	2
<b>2.0 - Introduction</b> .....	2
<b>3.0 - Release of easier programming languages</b> .....	2
3.1 Example of the library that GO provides .....	3
<b>4.0 - MapReduce</b> .....	3
4.1 Diagram of the MapReduce procedure .....	4
<b>5.0 - Modern Application Development</b> .....	4
5.1 Example of an application using microservices .....	5
<b>6.0 - Conclusion</b> .....	5
<b>7.0 – Bibliography</b> .....	6

Has programming become too easy?

## 1.0 - Position

Programmers used to write algorithms and implement data types; now they just glue existing components together.

## 2.0 - Introduction

Over the years, the process of developing software has rapidly changed from what it once was. From what started as machine code written in binary, soon progressed into high level languages such as java and C#, used for much more complex programming. What can be done now far exceeds what was possible before as technology is constantly evolving (High-level programming language, 2020). But as programming has become more understandable with more available resources than ever, has computer programming become too easy?

From an outsider's perspective it may seem that modern programmers only re use already written code or copy code from the internet, but I disagree with this perception. In a rapidly changing environment, it's critical that programming keeps pace with the needs of society, whether that may be handling massive amounts of data to support commerce or designing programs to suite the millions of different user requirements.

I will examine why easier programming languages with a large library are useful to programmers, and not just used to easily glue together to produce a program. I will discuss an approach companies are taking when developing modern software using a system called microservices, and an idea that may look like a lazy and easy approach but, is extremely beneficial to business.

## 3.0 - Release of easier programming languages

Easier and more comprehensive programming languages have been on the rise for a long time and very loved by the programming community (Rao, 2018), for obvious reasons. These languages include GO (Go (programming language), 2020) which was developed at google and is quickly becoming the favoured language for anything cloud related and has been described in an interview with the lead designer on 'thenewstack.io' as "the language that fits in a programmers head" (Cassel, 2019), and swift which is a backend language used exclusively by apple products with similar functionality and frameworks with other languages (Ariel Elkin, 2017). What these languages have in common is that they are high performance and are safe to use, yet also cost effective. This is due to how they are designed to be incredibly easy to use, through features such as portability, simplicity in code, easy to understand syntax, and a large standard library used to seek already written code.

Has programming become too easy?

### 3.1 Example of the library that GO provides

Name	Synopsis
archive	
tar	Package tar implements access to tar archives.
zip	Package zip provides support for reading and writing ZIP archives.
bufio	Package bufio implements buffered I/O. It wraps an io.Reader or io.Writer object, creating another object (Reader or Writer) that also implements the interface but provides buffering and some help for textual I/O.
builtin	Package builtin provides documentation for Go's predeclared identifiers.
bytes	Package bytes implements functions for the manipulation of byte slices.
compress	
bzip2	Package bzip2 implements bzip2 decompression.
flate	Package flate implements the DEFLATE compressed data format, described in RFC 1951.
gzip	Package gzip implements reading and writing of gzip format compressed files, as specified in RFC 1952.
lzv	Package lzv implements the Lempel-Ziv-Welch compressed data format, described in T. A. Welch, "A Technique for High-Performance Data Compression", Computer, 17(6) (June 1984), pp 8-19.
zlib	Package zlib implements reading and writing of zlib format compressed data, as specified in RFC 1950.
container	
heap	Package heap provides heap operations for any type that implements heap.Interface.
list	Package list implements a doubly linked list.
ring	Package ring implements operations on circular lists.
context	Package context defines the Context type, which carries deadlines, cancellation signals, and other request-scoped values across API boundaries and between processes.
crypto	Package crypto collects common cryptographic constants.
aes	Package aes implements AES encryption (formerly Rijndael), as defined in U.S. Federal Information Processing Standards Publication 197.



Easy to use programming languages are a great thing, but are they against progression in the programming world? It may look that way but in my opinion that is not the case, programming languages are desired to be easier for many reasons and being so helps allow the world of programming to constantly evolve. While reusing libraries of previously written code may seem lazy, it greatly speeds up the development of programs, there is no need to re-invent the wheel! Accessible libraries allow the programmers to not waste any time and go beyond their capabilities by giving them tools to take advantage of and learn from. Libraries contain well designed code that has already been tested and published with minimal bugs and using them allows the programmer to focus on solving problems instead of creating something which is already available, then spending time testing and debugging it (Hanselman, 2013). While it may seem that programmers only glue existing components together, a programmer still needs an extensive knowledge of programming and technology in order to successfully and efficiently develop software. Not only does easy to use languages benefit professionals with a list of advantages, but also aspiring programmers as it makes the profession so much more accessible. Easier syntax and understandable error messages provided by these new languages can only be a good thing as it helps the programmer understand what is happening, and where they are going wrong and I believe this is the way forward for programming.

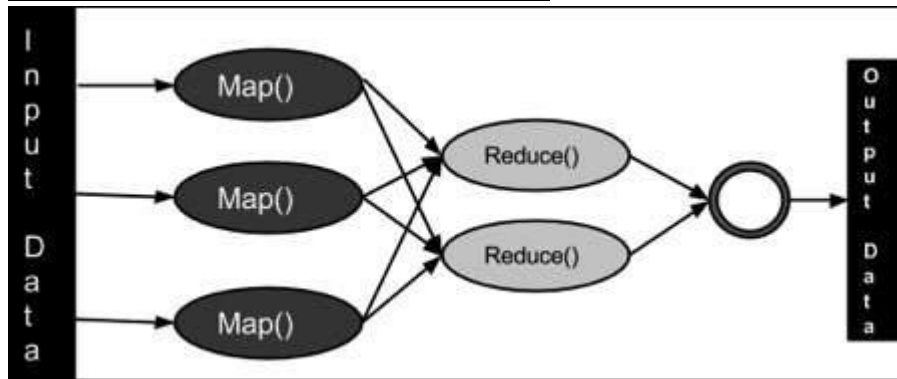
## 4.0 - MapReduce

While libraries are great for time saving, they can also be used with complex algorithms that deal with an extremely large amount of data, called big data. This algorithm is called MapReduce (MapReduce, 2020) and it is executed through library calls. The process of MapReduce takes a large amount of data as an input, then goes through a mapping procedure which changes it into a different set of data using mapping which gives each element a key, these are called tuples (Hadoop - MapReduce, 2019). All the elements in the initial data set are assigned keys and the output of the Map stage becomes the input to the Reduce stage. The MapReduce then goes through the process of reducing which takes the

Has programming become too easy?

set of tuples from the different map outputs and combines these together to create an even smaller set of tuples. This is done in parallel on a set of computers that work as one system over a network. This algorithm is useful due to its scalability, whether the amount of systems in the set is hundreds or thousands, this is done quite easily and the algorithm within the libraries can handle whatever the programmer needs it to. The output of the MapReduce is stored on a file system and can then be used and inspected.

#### 4.1 Diagram of the MapReduce procedure



Something like MapReduce has become a requirement as technology evolves and the number of users, and data, has significantly increased. It has a large number of uses and makes working with big data much easier than it used to be which is why it is gaining so much popularity. Some of these uses include letting businesses to identify which of their products earn them the most profits, how their advertising campaigns perform and what they should spend more money on, also weather predictions over a large location, and any other uses with large data sets. Before this was invented it was incredibly hard to process this data and would require an extremely educated group of computer scientists to figure this out, and incredibly powerful computers to process the large amount of data. Due to libraries and frameworks, an advanced team is not required to do this, in fact it is very easy and low cost to implement MapReduce. Using something like MapReduce is not just gluing components together, it allows programmers to exceed what was once possible and use these tools to advance technology. It gives less experienced programmers access to something that is very important due to the demand of programs required to work with big data. The amount of data held is constantly expanding meaning new solutions are needed and programming has changed to help accommodate the increase of users and data.

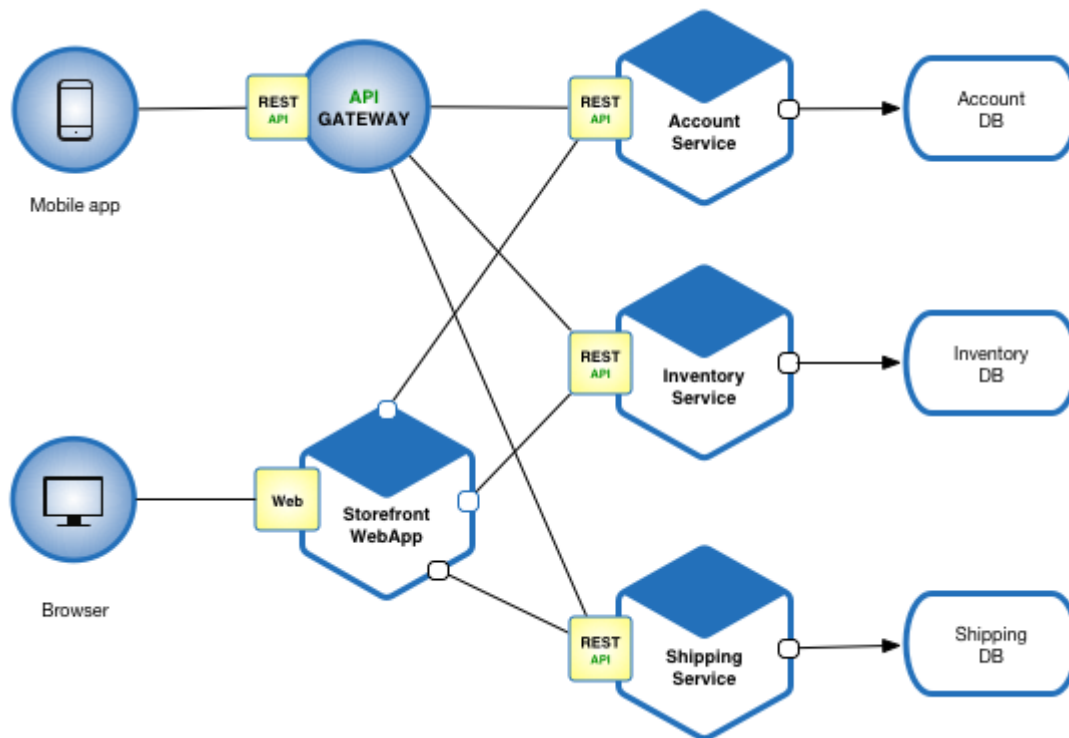
## 5.0 - Modern Application Development

While the number of users has increased, the range of devices of these users have also rapidly increased, along with many other requirement changes. This has caused a new approach needed when developing software, called modern application development. From a page published by Amazon (Modern Application Development, 2020) expressing the change in the development of applications we can find out what is now required in modern development. Amazon states that programs must now “scale quickly to potentially millions of users, have global availability, manage petabytes if not exabytes of data, and respond in

Has programming become too easy?

milliseconds.” Due to there being such a high demand for modern programs, everything needs to be done much faster, cheaper and with less risk. Using modern application development, companies focus on developing software in small packages with each their own function and allowing these packages to work together usually over a network, this is called a microservice (What are Microservices?, 2019). Microservices make implementing new features extremely easy with a much less risk and for much cheaper. Microservices are self-contained pieces of code, meaning if one part of the collection isn’t working correctly, the rest of the application can still work without it.

### 5.1 Example of an application using microservices



Each part of the collection doesn’t need to be coded in the same language, so any language can be picked meaning programmers aren’t constricted to one language they can use whatever will work best. Microservices are much smaller applications and do not have a large scope, allowing a team of people to each work on a separate part independently. This makes it easier for the programmer to implement but it is not a lazy approach to programming as it may look like the bare minimum of work is being done on an application, but this is not the case as it just provides so many benefits. It makes adding new features much safer, cheaper and can be scaled to however much pieces the programmer wants to implement. There is no need for a company to limit itself to the standard approach. Due to the new demands and requirements of users, things must now be released much faster in order to be successful and innovative as a company.

## 6.0 - Conclusion

In conclusion while it may seem that programmers just use someone else’s code to create programs, they need these to quickly implement already created features giving them the

Has programming become too easy?

time to focus on actual problems that have yet to be addressed. It allows them to evolve and create much more complex code without wasting time trying to build something that is available to them. The data that we deal with has also changed and algorithms are required to deal with such large data sets in a way that makes it incredibly easy and efficient, that doesn't require a full team of high level computer scientists to implement every time it is required. The process of developing software has changed and there are many individual needs ranging from different devices to instant response, causing a modern approach in order to develop software fast enough to suit everyone's needs. Microservices makes producing software a much quicker, more reliable and safer process helping companies quickly implement new features in order to keep up with the rapidly changing world of technology.

## **7.0 – Bibliography**

- Ariel Elkin, M. H. (2017, March 1). *Swift: Probably The Best Full-Stack Language in the World*. Retrieved from sitepoint: <https://www.sitepoint.com/swift-probably-the-best-full-stack-language-in-the-world/>
- Cassel, D. (2019, Feb 18). *GO, the Programming Language of the Cloud*. Retrieved from The new Stack: <https://thenewstack.io/go-the-programming-language-of-the-cloud/>
- Go (programming language)*. (2020, March 7). Retrieved from En.wikipedia.org.: [https://en.wikipedia.org/wiki/Go\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Go_(programming_language))
- Hadoop - MapReduce*. (2019, January 24). Retrieved from tutorialspoint: [https://www.tutorialspoint.com/hadoop/hadoop\\_mapreduce.htm](https://www.tutorialspoint.com/hadoop/hadoop_mapreduce.htm)
- Hanselman, S. (2013, August 23). *Am I really a developer or just a good googler?* Retrieved from hanselman: <https://www.hanselman.com/blog/AmIReallyADeveloperOrJustAGoodGoogler.aspx>
- High-level programming language*. (2020, January 13). Retrieved from wikipedia: [https://en.wikipedia.org/wiki/High-level\\_programming\\_language](https://en.wikipedia.org/wiki/High-level_programming_language)
- MapReduce*. (2020, February 25). Retrieved from Wikipedia: <https://en.wikipedia.org/wiki/MapReduce>
- Modern Application Development*. (2020, February 14). Retrieved from aws.amazon: [https://aws.amazon.com/modern-apps/?nc2=h\\_ql\\_sol\\_use\\_mad](https://aws.amazon.com/modern-apps/?nc2=h_ql_sol_use_mad)
- Rao, V. S. (2018, August 30). *Overview of Best Programming Languages For Cloud Computing*. Retrieved from technotification: <https://www.technotification.com/2018/08/programming-languages-cloud-computing.html>
- What are Microservices?* (2019, September 13). Retrieved from stackify: <https://stackify.com/what-are-microservices/>