Difficulties with Difficult Coding. Can Easy Coding create positive Social Economic impact?

By Mr Oh, an Electronics Chartered Engineer, member of Engineering Council of UK and a lecturer in so called Difficult Coding C++, Python.

Our lives are changing at an extremely fast pace today and along with that the process of learning has also accelerated at a similar alarming pace. Skills which were very specialized and exclusive a short while ago have actually become essential and critical living skills for today. For example, twenty years ago an average person will probably just know how to use common computer applications like word processors or email messaging system without really understanding how they work. Moving forward, ten years ago the requirement for computer knowledge of an average person has definitely increased significantly and just knowing how to use computers is no longer sufficient – the skills to understand how computer works and the basic construction and operations of computers have become a necessity. An average person, in addition to just knowing how to operate a computer correctly, also needs to perform tasks like simple troubleshooting of errors, configuring and installing applications and performing other simple maintenance tasks. Recently, the situation has again changed tremendously and probably in the very near future, an average person may have to know how to write programs or do "coding" to run on computers to perform specific tasks – just like nearly everyone knows how to drive a car or ride a motorcycle today.

Writing computer programs or "coding" will probably become the next critical skill for nearly everyone in the very near future – maybe not to write/develop complex and large applications but certainly to be able to understand and write common programs which are freely used in our daily lives. Most would have heard about !oT (Internet of Things) and how the internet can offer an electronic connectivity to almost any device and thus making them remotely manageable and controllable. Many things which were impossible years ago have now become a possibility because of IoT – for example you can remotely switch on/off your electrical appliances, control your air-conditioning/heater or monitor your CCTV cameras using just a simple smart phone. IoT has become very pervasive today but we will be even more dependent and attached to it in the future. An understanding of IoT and how it operates together with the skills to manage the IoT devices will become a common requirement for most people. Though IoT is mostly electronics, to enjoy the maximum benefits from IoT, a good understanding of coding or programming and the ability to write programs to manage the devices and systems will be a necessity. STEM (Science, Technology, Engineering & Mathematics) skills along with coding/programming skills have been greatly emphasized in recent years globally. Demand for programming/coding classes have escalated to unprecedented levels in recent years with many parents rushing to register their children to ensure that their children are prepared for the future.

Unfortunately, something is really missing though it may not be noticeable clearly to most. I am an IT professional for many years and programming/coding has been my professional core competency. Apart from being a practicing IT professional, I have also many years of lecturing experience in the field of electronics and programming/coding. When I look back at the past, I noticed that very few people actually had a genuine interest, understand programming/coding clearly and can competently apply their skills in this area. As an IT professional, I have come across many in the field who could not write good and efficient computer programs, with some even struggling to complete the simplest and shortest. As a lecturer, I also find that in most of my classes a very significant number of the students are facing difficulties and struggling to catch up with their lessons and could not competently program or write code. In my earlier years, I believed that programming or coding was just a field which is difficult for the beginner but after a while, once the student gets adjusted and comfortable, he/she should be able to do well. However, I was proven wrong later through experience – many were not able to improve even years later, probably because they fail even to grasp the fundamental concepts. The learning material and books are not easy to understand in most cases and the method of teaching and assessing programming and computing skills are not effective. In short, it appears to me that many are forced to learn coding or programming merely for the sake to seek employment even though it is not a field of interest to them nor something which they have a genuine interest for.

I may teach a programming subject like C++ and at the end of the semester (probably with between 14 to 16 weeks of lectures), students will sit for an exam where they have to memorize all they have learnt and write or code programs based on their memory. Definitely this is not an effective way to learn and develop but unfortunately this is the only option available. Programming and coding is difficult, uninteresting and even torturing to most (to quote some of my students/ex-students). Students are forced to learn and memorize programs and their syntax and there is no natural way to learn in a more intuitive and constructive manner. It is not like learning a new communication language like French or Mandarin where you can learn in incremental portions and immediately put to test what you have learnt by communicating with others you know the language well. If the student misses some important concepts, there is very little chance that he/she can recover and continue on to the more advanced topics. There was no possibility for the students to practice and learn coding in a coordinated manner in most cases. This is the journey of “**difficult coding**”. There were simply too many challenges and things to accomplish - remember/memorize language syntax, integrating together different programming concepts, limited intuitive examples to demonstrate concepts and a very uninteresting learning environment. Learning just coding is already a daunting task and what more if IoT and electronics is to be included? This will add new difficulties and complexities unthought of earlier. How then to convert “**difficult coding**” to “**easy coding**” ?

What is so magical about “easy coding” as represented by RunLinc/STEMSEL then? There is no magic at all in the product but it is merely a complete and comprehensive tool which will help accelerate the learning pace of students in learning programming and IoT, an integral part of STEM and also in digitalization and modern living. This “easy coding” product is able to introduce programming and IoT to students in a very simplified and practical manner quickly, without having students to build a strong foundation in coding first – thus it saves learning time tremendously. Students who would otherwise face enormous pressure and anxiety in trying to learn coding, and who unfortunately may also fail at the end after investing the resources, now have a much easier and structured way to learn, understand and apply coding skills effectively with “easy coding”. The difficult part of coding which involves the need to have a deep understanding of the syntax of programming languages and having to memorize syntax is taken away by the tool and instead the students can focus on developing their solutions first and have the tool worry about developing the framework and complex code to make the solution workable. A good analogy to this, though used in a different application, is a camera. Prior to the invention of cameras, artists have to depend very much on their drawing skills to capture an object, whether it is a still life item, a person or a scene, and to bring it to life. With the advent of cameras, the drawing skills are no longer critical and the artist can focus only on his creative skills to bring the object to life.

Finally, consider the implications of “easy coding” to complete the whole story. “Easy coding” is much more than just an accelerated and enjoyable method to learn IoT and coding. It simplifies real world adult problems and makes them comprehensible even to children, thus making children more responsible and understand the challenges they will be facing at a much younger age but without the assertion of unnecessary pressure to develop and learn faster and earlier. Through “easy coding” young children can understand and learn about global challenges like digitalization, green energy and related initiatives, environmental changes and sustainability better. All these are indirectly possible through the “easy coding” initiative – learning from the codes or programs of others (including those from adults and experts), opportunity to network with others and learn as well as try out their own ideas using the “easy coding” platform. Non-technical or computer savvy students and professionals who have ideas about using computers and coding to solve their challenges, but who do not have computing and coding skills, now have an easier alternative to test out their ideas and to evaluate the effectiveness. The ability to remove the myth that computers and coding is something beyond the reach of the average person except for the experts can be removed. Coding can become a basic fundamental skill for everyone just like driving, performing simple arithmetical calculations or communicating using everyday language. Imagine what this means – a lot of complicated or dangerous tasks can be simplified and automated, better educated and trained workforce in the future, improved quality of life and a more healthy and sustainable living environment. In summary, the benefits of “easy coding” are enormous – the world we live in will improve and global problems like poverty, pollution, global warming, sustainability and other related concerns can be effectively neutralized. World economy will show a significant improvement and the world will finally be a more pleasant place to live in.

By Mr Oh

Additional notes from STEM Professors

NO STEM. NO MONEY. NO SECURITY
STEM smart products need coding.

2 choices of coding: Difficult Coding (99.9% users) and Easy Coding

Students who use Easy Coding for 10 years from 10 to 20 years old to make smart IoT STEM products could enhance the chances to win 3 types of competitions:

1. STEM invention competitions with Social Impact
2. SCHOLARSHIP competitions.
3. JOB competitions.