1 Zakaj je programiranje pomembno ?

It’s been just over a year since I started my quest to learn how to code. My journey began while using a web –based interface to report data to the corporate office for a client of my employer located in another country. The information was due every Thursday. After compiling it all into MS Excel spreadsheets which were created to track the data, that data would then be transferred via re-keying into the web program. This task was tedious and seemed unnecessary. Often there was information that was required on the spreadsheet from other websites that would have missing, important, information. There had to be a better way to consolidate the process and put the website together so it could be the primary source of input, without worrying about what day of the week it had to be reported, and be able to have the site pull the information automatically from the other required websites. Realizing that there are already trained individuals in the field of coding and knowing that the entire process could be outsourced, it would have been easy to hire a third party, explain the concept to them, and have them start work immediately. I decided that learning to code is like learning a second language. Unless I could speak, read and comprehend in some manner there would always be an unnecessary communication gap. Thus began the first leg of my difficult journey to learn to code.

After researching various programs, and enjoying the creation story of Python, it seemed like the best place to start. The search was on for a free education site that would somehow overnight give me the understanding and tools needed to learn to code. Being one that learns in a visual manner, I knew that I needed videos to watch and follow along with to give me the ideal educational process. The first site I tried had that feature but was coming at the programming more from a math perspective. I got so caught up in not understanding the math processes that I couldn’t begin to understand the coding steps. This wasn’t working.

Off on the second leg of my journey to yet another site. This time I tried using a free app for IPad. I followed along with videos, check, that worked for my learning needs. Exercises were immediately after the videos and then at the end of each chapter there were additional exercises to work through. This idea was great but from the first video it felt as though a solid foundation of coding definitions and ideas were already needed. Going through the process was difficult and frustrating. The video explanations of the answers only caused more confusion and felt aimed at the idea that one already had a base idea of what was going on.

The third leg took me to trying to find resources where I could self-learn Python. That lasted all of two words being printed to the screen, time to move on.

The current leg of my journey brought me to Coursera. What seems to be working is I have a laptop with a dual monitor attached so I can watch the videos and code along as they play. I can use the app to watch the lectures, if needed, when not at home, and most importantly there is a textbook. Being able to print the chapter for each class has been the most beneficial feature. I hope that at the end of this class I have a solid beginning foundation to this new language, que es muy importante.

I wonder sometimes about this issue, “why is it so hard to learn programming?” due to my short experience in programming. I dare to rebuild the sentence and ask myself, “why is it so hard to learn a human language a foreign language?”, after all both are languages, both have reserved words, and both have norms and rules to build sentences, in order words grammar. One can also say using both of them “hello world”. Perhaps both are not very different.

They have more in common than I expected. The order, the syntax is keystone to make sense a sentence, on the other hand one has to choose the right word if one wants to **communicate successfully** and doesn’t want be misunderstood a similar situation takes place when someone wants to program, one has to choose the way to transmit the idea and put that idea in order to be understood by the computer.

Something that I have learnt studying both of them human languages and programming is “Don’t underestimate the small details”, they play an important role and should not be forgotten, unless the meaning can be changed or one can’t reach the expected answer, as well as colons and semi-colons are more than important.

As a beginner programmer I think one of the hardest things to understand is the different thinking paradigm needed to think in the logic way that the computer / language operates.
For example when I was first learning C I found it hard to understand why a statement such as
if (x == 1 or 2) would not work. The result wanted was that it would be TRUE when x ==1 or x ==2. My teacher had to explain to me that what I had written was x == (1 or 2).
1 or 2 could evaluate to 3 (if a bitwise OR) or it could evaluate to TRUE if a boolean OR. In either case it was NOT my intended result.
I had to rewrite as if ((x == 1) or (x == 2)) to get the required result. After this I used many () parentheses to ensure that the precedence of calculation was as I intended.
Another point that can cause confusion or a sense of feeling 'lost' for some students is that the feel they can not understand what is really going on 'under the hood' of the computer.
We write some text on the screen and somehow 'rate' 'hours' and 'pay' are able to store values, and values can be passed to functions which use different names. How does a function call 'work' ?
As an embedded C programmer it was only when I started to look at the assembly output from the compiler that I could understand how this all worked within the computer.
I could see the way values were copied to specific general registers prior to a function call. I could see the way the stack pointer was manipulated to reserve more space on the stack for temporary storage of local variables.
I could see the stack pointer being reset at the end of a function to free that area of stack. I could understand that a local variable will cease to exist when the function ends. For me this was important for my understanding, I felt lost programming for a world I could not understand before I knew all of this, but not every student needs this knowledge to feel comfortable programming.

In Python, as I began to study this great language recently I felt the same sense of confusion at first. But by experimenting at the Python command line I came to realize a bit more about how the Python interpreter is working.
I can imagine it as if every object (variable, class etc) is all referenced within Python by pointers, and that each time to create a new instance of an object the Python interpreter and OS work together to reserve some memory which is given an identity. By using the id() function you can see this identity value and understand a bit more deeply how values are passed around python by reference, and each object also has room in the memory allocated to it to store its attributes __type__ __name__ etc.
For some students this may be a necessary step to understanding how python is really working on their computer.

I first was introduced to programming when I was completing courses at Ohlone College for an Electronics A.A.S Degree. One course was named "Introduction to Micro-Controllers" where I had to program a (8086) microprocessor using Assembly Language. Assembly Language is a form of code that controls the processor. My first program was turning LED's(little light) on and off, but to me this was a big deal. I was hooked and I wanted to know more about controlling this processor to do more sophisticated things. I approached the professor and ask what other courses I could take to continue learning programming and he directed me to a course that taught a computer programming language called "Basic". This was back in the early 90's when Basic was still used. I took (1) semester and after thst I began using programming in my electronic circuits. I could control my circuits using a program I wrote.

First, I started with basic programs to turn LED's on and off like in my micro-controller course and as I grained more knowledge in "Basic" programming I wanted to see how far I could take this. I used "Basic" to control motors, interface with electronic test equipment, and to control more elaborate circuits. Next I wanted to build programs that had buttons, switches, and lights, so I took (2) Visual Basic courses so I could create GUI's, and finally I took (2) courses of C++ because I was told this languages is what I need to know to take my projects to the next level.

After all that I had to put programming on the back burner, and so I have been out of the programming as a hobby for a number of years now, but as I look back I wish I would have tried programming as a profession.

I work at Google and I'm around a lot of programmer (coder, some of the programmer use this word) and being around these coders got me interested in getting back into programming. I heard of "Python" though a user group at Google and how easy it was to learn. Yes, Google uses Python one of many programming languages they use and that's why I'm taking this course. Google has an internal course but when I checked into it the course is for a more advanced programmer and my programming skills are very rusty. I was told by the Google instructor that they offer this course to try and recruit new potential programmers because they are always in need of good programmers.

Yes, in the beginning learning programming was slow going, but it was so amazing to me what I could do with it. I guess the reason I didn't quick was I wanted to use it in my electronic projects. That might be a tool to apply
to keep motivated when you want to quick. Have a goal. Maybe developing your own web page, write your own software app that does something that no other app does, or maybe a new career(Google is always looking I'm told). Oh yea, great place to work (lots of free food).

Programming is the future and the future is here. Computer programs are everywhere in are appliances, cars, and in almost every industry. Programs are what give a robot instructions to do a task, and what about home automation
"Smart Homes" without programs it's a dumb home .

I want to learn this language and maybe take some refresher courses and try to transfer into a programming job at Google. This is my goal this time around.

Happy coding!!

Not English native…

Why is programming so difficult? Because IT IS!

First, you have to change the way of thinking – you have to adapt your way of thinking to the computer's "way of thinking". It is usually difficult. It is also hard to understand why you cannot obtain the result that you want with your "perfect" program. You "told" it to do something and what is your computer doing? Something different or… nothing, keeps giving you error messages… There is always a difference between what you think you "told" to your computer with your program and what you actually "told" it to do.

Second, every book related to programming looks at least intimidating; either is really big - lot of pages - either uses strange characters or words and concepts that are hard to understand.

Third, there are so many rules to remember… data types and operations, variables and constants, and functions, and procedures, and so on… And you have to spend hours to practice and debug your code.

And that's not all. When you think you are familiar with a programming language, you have to switch to another version or to a brand new language, because, of obviously, your new company uses another language, or some new trend appeared or a new version of the program is available.

***

There is no easy way to learn programming. But there are things that can make your learning easier. You have to have an interest in computers. You have to understand that you'll need time, need to practice, need perseverance in your learning. Having friends with similar interests can make you learn easier.

When your first program is running and gives you the expected result - well, you'll feel great!

You can always stop your learning - it is easy to abandon, but if you choose not to, the reward is great.

So, why learn programming?

Because it is fun. Because it organizes your way of thinking. Because you can make money.

;-)

Why I think it is so difficult for many people to learn programming? Here are a few reasons:

- It takes time to harness that skill.
- It is very easy to make syntax errors .
- It is difficult to make a useful program.
- You do not know what language to choose.
- It is difficult to make a "Hello World" program for the first time in your computer.
- Not many friends of yours have the skills to help you.
- It a different way of thinking (in comparison with many other studies).
- The resources are primarily in english.
- Most people do not know how to use internet for answers.
- Slow internet connection.

I think one of the most difficult challenges in learning to program is making the transition from every day natural language and punctuation symbols to the world of programming where those symbols have specific and precise meanings and need to be in the right place at the right time. When first looking at a chunk of code, I recognize the symbols like a colon, : , or a forward slash, //, but I don't know what they mean. It looks like it has no meaning and I don't know how to unlock the information that is being displayed. One of the first courses I took simply had us do exercises in coding but without understanding the why or how the code worked. It was like practicing the piano scales without learning the notes or how they are related to one another. The practice did teach me the pattern of coding but it was frustrating to not know what it all meant.

Another factor that makes programming difficult to learn is, of course, that in the beginning, error messages are a constant reminder that your skill level is at the beginner level. Pushing through to getting a correct piece of code to work is satisfying, but it is a challenge to persevere at the beginning.

I don't find learning programming difficult at all. But then again, I have Asperger's Syndrome.

This is my first time learning a programming language in a course structrure. It comes easy to me because so far there are no multiple-meaning statements, there is a logical and defined set of rules, and if you want something, you have to tell the machine the exact command for that.

No misinterpretations, no hiding your intentions, no saying something completely different than what you are thinking and leaving the interpreter to puzzle out your body language and facial gestures.

Simply that, simple ideas conveyed through simple language. Precisely what I'm trying to do with this very short essay.

But if you really want to know why programming is difficult for most people, I suppose you just have to read the previous paragraphs inverting the ideas, just like you would with logical operators.

2 Zakaj je programiranje pomembno?

Nowadays, programming is absolutly present in this technological world. Programming is everywhere, in our smartphones, in the computers, in smartTVs or in the ATMs. So, I think it would be a good idea to let children learn programing just as another subject at school. An for some of them, it would be a hateful subject, for other just another and for the rest a lovely subject. Just as maths, geography or philosophy, for example. Programming is a way of resolving logical problems, so it would be another tool for children to grow up. Then, they could choose if they like it or not and decide if they wanted to learn more in a degree or so, but at least they would have the basis to understand how the things that sorround them work.

Programming is essential for everybody like **reading** . I think it is more than that . As i heard in TED talks and other presentations, it is very important for every user of an computer to learn programming.

https://youtu.be/xfBWk4nw440

This talk inspired me to learn programming.

Programming has become an integral part of our society today. All the electronic gadgets that we use to make our lives what they are today depend on it. It is found in household appliances such as microwaves, stoves, fridges, washing machines and even toasters. Medical equipment from simple devices to measure temperature to the complex Magnetic Resonance Imaging (MRI) scanners all depend on programming. They rely on embedded systems[1], which are computer systems with a dedicated function within a larger mechanical or electrical system. Programming controls the various mechanical systems and electrical systems in vehicles, trains, planes, space ships and satellites. We are living in an electronic age[2].

Without consumer electronics including mp3 players, mobile phones, videogame consoles, digital cameras, and GPS receivers[1], life would be very different. It would be far less comfortable and impossible to do many things we have accepted as common place. Programming is therefore very important.

Most of us are using items which depend on programming daily. Does this mean that everyone needs to learn programming? No! These systems are designed with user interfaces which make use easy and intuitive. So what would the benefits of learning to program be?

Programming is a tool which can be used in problem solving. Learning a programming language is not difficult. The rules are far more basic than mathematics which is offered in schools from an early age. Learning to solve problems within the framework of the computer or microprocessor comes with practise. We all agree, that offering children balance from an early age is beneficial. We encourage sport because it has many spinoffs other than just physical fitness. It improves perseverance, your scores on tests[3], grades[3], sleep [3], energy and your mood[3].

Programming will also be beneficial in other areas.

Programming is creative. There are so many different and exciting ways of solving one problem. Creativity has many benefits:

· Creating helps make people happier, less anxious, more resilient and better equipped to problem-solve in the face of hardship[4].

· Confucius once said that if you choose a job you love, you will never work a day in your life. Confucius must have known then what science now confirms: Passion protects us physiologically, allowing us to work longer and harder than we would be able to toiling away at a job we hate[4].

Programming highlights how repetitive jobs can be automated, so instead of spending hours collecting, analysing, processing and reformatting data in order to present something useful, a simple program can be written to perform the same task.

Becoming skilled at programming would entail the same as becoming skilled in any other activity. Firstly an individual needs exposure to programming. Then they need a place to practise the skill and to be shown how it can be used to improve their own personal world. And then, to become successful, they need to fall in love with it!

Given the opportunity to learn programming from an early age will give children an opportunity to exercise their creativity, to improve their problem solving skills and learn to think out of the box. The benefits in later life would be, that even if it is not something they develop a passion for, they will know about how it works and be able to incorporate the solutions it offers into their own environments, even if they will be tasking a programmer to write something for them.

Everyone can benefit from learning programming!

I'd like to start my essay with the quote: "Everybody in this country should learn how to program a computer... because it teaches you how to think." These are the words of Steve Jobs, and I completely agree with him.

Programming is incredibly empowering skill to learn. We are already surrounded by computers, and their number will only increase next years in all areas of our activity. So it's important for everyone to know how to manipulate it. Furthermore, Drew Houston, founder and CEO of Dropbox, said that programming is the closest thing we have to the secret power, and I believe that he is right.

Computational thinking is a skill that everyone should learn. Even if you never become a professional software engineer, you will benefit from knowing how to think this way. It will help you understand and master technology of all sorts and solve problems in almost any discipline.

This is not primarily about equipping everybody to work as software engineers. It is about promoting computational thinking. Computational thinking is how software engineers solve problems. It combines mathematics, logic and algorithms, and teaches you a new way to think about the world.

Will every job in the future involve programming? No. But it is still crucial that every child learns how to code.

https://www.youtube.com/watch?v=Dv7gLpW91DM
http://www.theguardian.com/technology/2014/feb/07/year-of-code-dan-crow-songkick

I guess to learn programming is more important than ever in our world. I wish it would had been taught in the primary school, at least the basics properly. I feel the pressure in our daily life while we use the internet for everything and the useful applications appear on every devices (smartphones, tablets and even game consoles, smart watches) that it is very necessary to know about one or even more programming languages and the way how we use it. I want to feel that it is definitely not only for men! I am a woman with a genuine interest for information technology and services and I would love to be part of the IT community proudly as a female programmer in the near future. I would love to know much more about the background of all the good things and I believe everyone has got this kind of interest, to know more and be able to use the developed products better than ever.

If you know how to program then you can do a lot of things with that skill. Maths is the basic of everything. You can go to any kind of direction. You can improve your knowledge with other skills like design more easily. For example, you can develop websites which is a very interesting job and I wish to do that for a living.

If you think of the computerized world then you can say that we, people are obsessed with digital technologies. We want most of the manual things to be digital and quick, to help our lives. The main element of every digital devices is programming. People are developing software applications to do this and that and whatever we want to see, appear on the screen immediately. We, people buy a lot of digital stuff, mainly as a Christmas present. It shows that we are quite interested in this sort of things and we really enjoy using them because it makes our life a lot easier.

I am pretty sure that young people (and even me too) might learn programming as a subject in school as basic knowledge - which could be upgraded to a higher level - because they are very curious, open-minded, gadget lovers and keep using the internet, the new information technology devices, we can say, the most of the time in their life.

As the computer world keeps changing year by year we also should be up-to-date about what is going on and what are the new developments, which new programming languages were born and which one is rather important to learn if you would love to get into the IT sector as a career changer. I truly should make serious steps in the education sector, so people could gain good or high standard of information technology skills if they want and feel more comfortable working on computer and any devices anytime and anywhere needed.

"Everybody in this country should learn how to program a computer ... because it teaches you how to think" - Steve Jobs

Program was always intimidating for me, the boolean logic has always been a confused interpretation for me, but with this course, I'm having a good way to learn and committing fewer errors in boolean, now I know how the computer works, I know that 1 and 0 can be numeric or logical: And, Or, No, decimal, hexadecimal, bit, bytes ...This works, it teaches you to think and to transmit that thought to run it !

I agree with Zach Sims, one of the founders of Codeacademy, he says : "Programming has multidisciplinar effect, wich involves several areas and understand it, make peoples better qualified to solve problems, deal with challenges and obstacles, these skills are important for all life.

According a video I watched about programming, 1 in 10 children doesn't know what is to be a programmer, (2013 video), and no one chose this career, this shows that instituitions are not prepared (this happened to me) and this demand a significant delay in the future, not only in IT, but also in the character of people, character that I could prove every time I locked in a programming problem: these peoples are team players, have time to help you, share knowledge contributing to a global and cultural knowledge which is independent of geographical location.

I say: "Hello beautiful world !"

3 Singularnost

Unique Human Intelligence

The belief that computer intelligence will surpass human intelligence is both exciting and scary. Imagine if artificial intelligence could improve our quality of life. Perhaps robots and machines could work with us to solve global warming problems, cure diseases or even go to work and make money for us! On the other hand, imagine if the new race of artificial intelligence schemes against 'humans' causing destruction and war. The capabilities are endless and so are the number of movies that have tried to visually predict this moment of 'singularity'.

In this short essay I will share my opinion, that artificial intelligence never will never exactly match or exceed our unique human intelligence. My main reason for believing that our intelligence will not be surpassed or exceeded is that we cannot directly compare human intelligence to artificial intelligence. Artificial intelligence may mimic human intelligence, but I believe that it must be different. Unique forms of intelligence have pros and cons, and this is seen in other organisms on our planet.

How do we fairly compare the intelligence of a cockroach and a human? Or a fish and a lion? These creatures have occupied their own respective niches on the planet successfully. They are able to thrive in their environment the best, because they have evolved over time to exploit a particular environment. For example, a lion may be much smarter than a fish in the African savannah, but it would not survive under the sea. A cockroach might seem pretty dumb compared to a human, but a cockroach will likely survive longer in a natural disaster where it can scrounge for tiny scraps, when a human would starve to death.

I think that if artificial intelligence reaches a point of singularity, humans would be able to use our unique human intelligence to survive separately from machines and robots. We would learn ways to diverge from artificial intelligence. We would evolve and occupy different niches. I think that the fundamental rules of biology would proceed and be applied. We would be in competition to survive, as if artificial intelligence were another type of animal to share the planet. It is so hard to tell, but I truly believe we would find ways to use our 'human' smarts in ways that artificial intelligence never could. Perhaps ironically, our unique ability to feel things and experience love would help our survival. Or perhaps our unique human desires and interests would have us happily living in a much different niche than our 'robot friends'.

The technological singularity is the hypothesis that accelerating progress in technologies will cause a runaway effect wherein artificial intelligence will exceed human intellectual capacity and control, thus radically changing civilization in an event called the singularity [1]. But how likely is this to occur?

First of, the U.S. Bureau of Labour Statistics predicts that by 2018 there will be 1.4 million open technology jobs in the United States, and at the current rate of students graduating with degrees in computer science, around 60 percent will be filled [2]. If there continues to be this deficit, caused by a lack of programming education in schools, the chance of technology superseding us will be a long way off. However, for arguments sake, let’s say the gap is suddenly closed, and technology gets everything it needs for us to reach its full potential. Will artificial intelligence be our downfall?

Artificial intelligence is a complicated subject, mainly because ‘intelligence’ in itself is complicated. Are we considering the capability to do complex things ‘intelligence’ or is it not until machines become self-aware that we consider them potentially dangerous? Humans are not born self-aware. It isn’t until a child is 18-24 months old that it begins to become aware of itself, and the consequences of its actions with relation to its surroundings [3]. This development requires a lot of complicated wiring and re-wiring in the human brain, and is something extremely difficult to mirror with technology. It is perhaps not impossible, but still an advancement that is a VERY long way off, and would probably require a biological computer system. The chances of our coffee machines deciding they’ve had enough of the human race and turning us into batteries is small.

However, if we choose to think of AI as the ability of a machine to be smarter than us, then in many ways it’s already happening. Computers are capable of running complex programmes in the blink of an eye. If you asked a human and a computer to calculate pi to a thousand digits, the computer would win, but that doesn’t mean then end of humanity. In actuality, the ability of computers to do complicated things we can’t is beneficial. They can run simulations on the effectiveness of drugs, monitor our environment to help keep us safe, and they can reduce the trauma of doing tax returns.

So in conclusion; will artificial intelligence exceed human intellectual capacity and control? Capacity, yes, and is already doable. Control, no...at least for the foreseeable future.

I believe that the singularity will come to our life in the future, and it will change our life, make our life more colorful, but technology won't become the master of the human race, because machines have no emotion, no evolution, it can't be stronger without human, There are many many words I want to say, like Artificial Intellegence, Machine Learning and other related topics, and python is playing a key role in these fields, but y english is a little bit poor, I just don't know how to describe them more detailedly.