**Reading comprehension task 1**

**8 Bits of Advice for New Programmers**

There’re many Articles related to [Top Programming Languages](https://codinginfinite.com/top-programming-languages-2020-stats-surveys/) & [Programing for Beginners](https://codinginfinite.com/coding-for-beginners/), but Today I’m going to tell you my 8 bits of advice for new programmers who just started out. This is the stuff that they won’t teach you at school.

I’m not going to talk about any programming language in particular, because the real root of being a great programmer is understanding the structure & the nature of the Code.

Learning programming is not all about syntax which can be learnt in days.

So, Let’s get started.

1. **Don’t make your life harder for yourself**

I think should learn to walk before you run. The first thing that every new programmer should learn is the basics of programming like variables, arithmetic operations & then follow the sequence, arrays, functions extending with structure & classes.

Then you should think about more advanced stuff like object-oriented programming concepts Abstraction, Encapsulation, Inheritance, Polymorphism. Going further, still talking about template chapter pre-processor stuff & if you want to go deep then you have to learn libraries, you’re gonna have to learn protocols. Don’t jump into the deep end. Make sure you have a clear concept of basics first. You won’t be careful about then what language you’re using. You understand how the processor handling the data & understand how these tool, CPU graphics can be leverage to your advantage.

So, don’t make the first thing you write you ultimate ambition start by doing the little program, learn basics, don’t set your expectation too high & remember you’re still a beginner. it may really be tempting to I’m going to write a computer gone or I’m going to create a massive complicated web-structure for my website. Be a bit more reason why not start off writing some console application simply, really simple stuff enter your name & if does funny things with the organization of characters or solve the athematic is an extended example for really staring out.

1. **Choose the Right Tools**

Choosing the right tool goes Harding in hard with the language you want to use but you just started out. Don’t start with the really complicated compiler chains, make tips stick to the rapid application development fully integrated stuff. You want tools which show you the complier information to see what it’s doing easy to debug your goals free development environment online but I recommend to stay away from the one that required you to compile then yourself. Start out with something really tightly integrated. Final a nice extendable that just click & run.

Most important whether the tool you choose make sure you can run your program within seconds that finishing the last line of code

1. **Keep it fun**

Once you adjust to the ground & you familiar with basics, try & write the program once you want to write but understand you’re not going to write it on gunpoint. Make coding fun for you to master it.

If you want to write games stuff with the small level games like puzzle, that’s be really good for teaching you the algorithm, thinking about data if you in to music stuff try & load a music file & see if you can understand where the file is structure then add some crazy efforts to it by playing it with the math.

If you’re into sports as a hobby, why not write code some application that keeps tracking the score.

Something you might actually consider using yourself & enjoy doing it.



1. **Practice**

When I was learning to code, the best way to practice is to create a whole program dedicated to learning one function. When I was learning loops, my program would contain for-loops until I learnt the topic clearly.

As your skillset developed, you got the whole library for a little program, which you can label & always use them for reference. But remembered it takes time, efforts & it takes practice. Don’t try & write a program at the start which has every single feature of the language.

Keep these little practices program as simple & discrete. They only do one thing & the one thing is to teach you how to code.

1. **Never Copy & Paste**

You just starting the worst thing that you to going online & copy the code & paste it. That’s very useful takes on but you just starting out get used to syntax by trying it out. Get used to writing concise code rather than reading robust code.

Get read into writing readable code getting in the habit of goal variable having the convention. You won’t be able to do any of this. If what you doing is copy & pasting from other people write yourself even if you copying it from another website. Trust me it’s the quickest way to learn.



1. **Don’t Listen to the People on the Internet**

Internet Forum has a particular type of people even though the intent can be a really good source of information. These are always people online who want to show how good they are. They are very arrogant programmers out So whenever a guy starting to help out you the first thing they ask you what you’re doing it wrong. You should be doing it like this, this that not correct, don’t want to use that technique. How is this helpful for somebody who is just starting out?

Don’t get me wrong there’s a lot of good advice to be found on the intent of avoiding asking people for help. But just identify that these are people out there who see you as a threat. Don’t let those rumors ruined your day by making your little programs really over complicated. It only leads you to frustration.

1. **Break the Rules**

You need to break the rules, to use global variables all over the place, use unsafe system calls, don’t check all of your integers to make sure they are in range. this is the stuff they don’t advise you to do in school but you know it doesn’t matter–you are learning, you are not gonna cause any harm, you are not gonna break a computer mostly. The computer doesn’t have any feeling you’re not gonna upset him & most of all your code isn’t been reviewed by somebody on the internet.

Do whatever you need to do to get the job done. it’s more satisfying & more rewarding & you want to keep up with it. Over time you can learn good practice & expected standards as well. Don’t swim with all the industry standards at the beginning. that will come with time.

Taken from:

**Answer the following questions:**

1. the real root of being a great programmer is understanding the structure and the nature of the Code.

**True**

**False**

1. A new programmer shouldn`t learn the basics of programming like variables, arithmetic operations & then follow the sequence, arrays, functions extending with structure & classes.

**True**

**False**

1. **Programming shouldn´t be boring.**

**True**

**False**

1. **According to the text, practice:**
2. **is not important.**
3. **is less important than theory.**
4. **is really important to learn how to program.**
5. **According to the author, copy and paste:**
6. **Is something a programmer should do.**
7. **Is something a programmer shouldn´t do.**
8. **Is something a programmer must do.**

**Based on the reading complete the chart:**

|  |  |  |  |
| --- | --- | --- | --- |
| **A programmer should** | **A programmer shouldn`t**  | **A programmer must**  | **A programmer mustn`t**  |