Dan Marvin Geronimo  
PROGLAN  
  
**10 Effective Ways to Become a Good Programmer**

1. **Work on Basics  
   -** You can’t have a thousand pesos without one peso. Every advanced system starts from the fundamentals.
2. **Start putting question tags (how, what) with every set of code you write  
   -** Similar to how writing notes for yourself during a math class works, simple explanations and question with lines of code go a long way when having to review or help someone who doesn’t understand, be it another person or yourself.
3. **-**By helping others, you get to see their mistakes and learn from those said mistakes, as well as their techniques when they code.
4. **Write simple, understandable but logical code  
   -**Writing complicated code on a project being worked on by multiple people helps no one but yourself, so be considerate for other people you’ll be working with by simplifying it as much as possible.
5. **Spend more time in analyzing the problem, you’ll need less time to fix it  
   -**A lot of people like to code/solve problems through *brute force* where they just go through every way of solving the problem until they do. But programmers nowadays don’t just code, they analyze as well.
6. **Be the first to analyze and review your code  
   -**Double checking and analysis is key. Especially with your own work as you may find yourself blind to your own mistakes quite often.
7. **Don’t dismay yourself by looking at changing technology world  
   -**Technology is constantly changing, as a result, more and more languages are being created/changed. But that doesn’t have to be a bad thing, because as more languages come, there’ll be more chances to learn.
8. **Work-arounds don’t work for longer time  
   -**Work-arounds are like how cheats/shortcuts work, they work right now but it won’t work all the time. Learning how to do it the proper way is the best way to future proof.
9. **Read documentation  
   -**A lot of people actually dislike reading documentation because of how intimidating it is, but once you start dealing with high-spec projects, documentation becomes your best friend, so why not be better acquainted as early as you can get?
10. **You can learn from others code as well  
    -**Again, learning from others minimizes your mistakes and broadens your own scope for coding.

**Two parts I actually implement are:**  
  
- **Start putting question tags (how, what) with every set of code you write**- **You learn more by helping others**I’ve been implementing these two especially since working on my thesis since last term, helping my thesis mates with modules they have difficulty doing and writing small notes and explanations on how blocks of code work in our project.