

Re: How do I start?

Source: <http://coding.derkeiler.com/Archive/General/comp.programming/2007-07/msg00427.html>

- *From:* Mike <m.fee@xxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 30 Jul 2007 11:29:18 +1200
-

In article <nrOdnTejD4SfdTHbRVnygwA@xxxxxx>, rjh@xxxxxxxxxxxxxxxxxxxx says...

Rui Maciel said:

Richard Heathfield wrote:

Alas, that is how too many people "get into computer programming" nowadays, which is perhaps why so many people are so bad at it.

For some people that is exactly the only way they can "get into computer programming". What's wrong with that?

Let's reiterate the advice on which I was commenting, shall we? (a) pick a language (b) get a compiler for it (c) grab a tutorial off the Web.

That strategy offers no guidance on how to pick a language that is appropriate to the mindset of the person choosing it. If they choose a language that turns out to be incomprehensible to them, maybe we'll lose a potentially excellent programmer because they are turned off at Stage 1. (b) is neither here nor there, really, so let's move on to (c). The quality of Web tutorials is highly variable, and again the newbie is not in a good position to judge quality.

So the advice can and does lead to frustration, inefficiency, and the spread of poor programming practice.

That is a bit precious Richard. All he wants to do is "get into computer programming" – it is not as if he will be writing Air Traffic Controller software within the next few months. I thought the advice was perfectly reasonable, analogous to being told, if you asked how to "get into painting", that you should a) pick a media, b) buy some cheap

Re: How do I start?

paints, and c) buy, borrow or steal a "How To..." book. In fact the only thing I thought that was missing from the advice was: d) experiment with the language and start writing programs. If he decides to make a career out of it then he will probably either a) undergo formal training or b) learn enough basics to train on the job. In either case, the quality, correctness, and robustness of the code he produces in the longer term will be more closely influenced by his intellect, work ethic, and working environment than by the history of how he first started programming.

Maybe the poster should have suggested pick a relatively popular language for his first exposure to programming but, lets face it, if he found any of c or pascal or lisp or forth or java or haskell or whatever "incomprehensible", he would probably not have the appropriate intellectual bent to become an excellent programmer in any language.

Mike

.