

## Texts: Real ...

- Software Engineers Don't Read Dumps
- Real Programmers Don't Write Specs
- Real Computer Scientists Don't Write Code

### Real Software Engineers Don't Read Dumps

Real software engineers don't read dumps. They never generate them, and on the rare occasions that they come across them, they are vaguely amused.

Real software engineers don't comment their code. The identifiers are so mnemonic they don't have to.

Real software engineers don't write applications programs, they implement algorithms. If someone has an application that the algorithm might help with, that's nice. Don't ask them to write the user interface, though.

Real software engineers eat quiche.

If it doesn't have recursive function calls, real software engineers don't program in it.

Real software engineers don't program in assembler. They become queasy at the very thought.

Real software engineers don't debug programs, they verify correctness. This process doesn't necessarily involve executing anything on a computer, except perhaps a Correctness Verification Aid package.

Real software engineers like C's structured constructs, but they are suspicious of it because they have heard that it lets you get "close to the machine."

Real software engineers play tennis. In general, they don't like any sport that involves getting hot and sweaty and gross when out of range of a shower. (Thus mountain climbing is Right Out.) They will occasionally wear their tennis togs to work, but only on very sunny days.

Real software engineers admire PASCAL for its discipline and Spartan purity, but they find it difficult to actually program in. They don't tell this to their friends, because they are afraid it means that they are somehow Unworthy.

Real software engineers work from 9 to 5, because that is the way the job is described in the formal spec. Working late would feel like using an undocumented external procedure.

Real software engineers write in languages that have not actually been implemented for any machine, and for which only the formal spec (in BNF) is available. This keeps them from having to take any machine dependencies into account. Machine dependencies make real software engineers very uneasy.

Real software engineers don't write in ADA, because the standards bodies have not quite decided on a formal spec yet.

Real software engineers like writing their own compilers, preferably in PROLOG (they also like writing them in unimplemented languages, but it turns out to be difficult to actually RUN these).

Real software engineers regret the existence of COBOL, FORTRAN and BASIC. PL/1 is getting there, but it is not nearly disciplined enough; far too much built in function.

Real software engineers aren't too happy about the existence of users, either. Users always seem to have the wrong idea about what the implementation and verification of algorithms is all about.

Real software engineers don't like the idea of some inexplicable and greasy hardware several aisles away that may stop working at any moment. They have a great distrust of hardware people, and wish that systems could be virtual at ALL levels. They would like personal computers (you know no one's going to trip over something and kill your DFA in mid-transit), except that they need 8 megabytes to run their Correctness Verification Aid packages.

Real software engineers think better while playing WFF 'N' PROOF.

### Real Programmers Don't Write Specs

Real Programmers don't write specs -- users should consider themselves lucky to get any programs at all and take what they get.

Real Programmers don't comment their code. If it was hard to write, it should be hard to understand.

Real Programmers don't write application programs; they program right down on the bare metal. Application programming is for feebs who can't do systems programming.

Real Programmers don't eat quiche. In fact, real programmers don't

know how to SPELL quiche. They eat Twinkies, and Szechwan food.

Real Programmers don't write in COBOL. COBOL is for wimpy applications programmers.

Real Programmers' programs never work right the first time. But if you throw them on the machine they can be patched into working in "only a few" 30-hour debugging sessions.

Real Programmers don't write in FORTRAN. FORTRAN is for pipe stress freaks and crystallography weenies.

Real Programmers never work 9 to 5. If any real programmers are around at 9 AM, it's because they were up all night.

Real Programmers don't write in BASIC. Actually, no programmers write in BASIC, after the age of 12.

Real Programmers don't write in PL/I. PL/I is for programmers who can't decide whether to write in COBOL or FORTRAN.

Real Programmers don't play tennis, or any other sport that requires you to change clothes. Mountain climbing is OK, and real programmers wear their climbing boots to work in case a mountain should suddenly spring up in the middle of the machine room.

Real Programmers don't document. Documentation is for simps who can't read the listings or the object deck.

Real Programmers don't write in PASCAL, or BLISS, or ADA, or any of those pinko computer science languages. Strong typing is for people with weak memories.

#### Real Computer Scientists Don't Write Code

Real computer scientists don't write code. They occasionally tinker with "programming systems", but those are so high level that they hardly count (and rarely count accurately; precision is for applications.)

Real computer scientists don't comment their code. The identifiers are so long they can't afford the disk space.

Real computer scientists don't write the user interfaces, they merely argue over what they should look like.

Real computer scientists don't eat quiche. They shun Schezuan food since the hackers discovered it. Many real computer scientists consider eating an implementation detail. (Others break down and eat with the hackers, but only if they can have ice cream for desert.)

If it doesn't have a programming environment complete with interactive debugger, structure editor and extensive cross module type checking, real computer scientists won't be seen tinkering with it. They may have to use it to balance their checkbooks, as their own systems can't.

Real computer scientists don't program in assembler. They don't write in anything less portable than a number two pencil.

Real computer scientists don't debug programs, they dynamically modify them. This is safer, since no one has invented a way to do anything dynamic to FORTRAN, COBOL or BASIC.

Real computer scientists like C's structured constructs, but they are suspicious of it because its compiled. (Only Batch freaks and efficiency weirdos bother with compilers, they're soooo un-dynamic.)

Real computer scientists play go. They have nothing against the concept of mountain climbing, but the actual climbing is an implementation detail best left to programmers.

Real computer scientists admire ADA for its overwhelming aesthetic value, but they find it difficult to actually program in, as it is much too large to implement. Most Computer scientists don't notice this because they are still arguing over what else to add to ADA.

Real computer scientists work from 5 pm to 9 am because that's the only time they can get the 8 megabytes of main memory they need to edit specs. (Real work starts around 2 am when enough MIPS are free for their dynamic systems.) Real computer scientists find it hard to share 308ls when they are doing "REAL" work.

Real computer scientists only write specs for languages that might run on future hardware. Nobody trusts them to write specs for anything homo sapiens will ever be able to fit on a single planet.

Real computer scientists like planning their own environments to use bit mapped graphics. Bit mapped graphics is great because no one can afford it, so their systems can be experimental.

Real computer scientists regret the existence of PL/I, PASCAL and LISP. ADA is getting there, but it is still allows people to make mistakes.

Real computer scientists love the concept of users. Users are always real impressed by the stuff computer scientists are talking about;

it sure sounds better than the stuff they are being forced to use now.

Real computer scientists despise the idea of actual hardware. Hardware has limitations, software doesn't. It's a real shame that Turing machines are so poor at I/O.

Real computer scientists love conventions. No one is expected to lug a 386 attached to a bit map screen to a convention, so no one will ever know how slow their systems run.

—  
return to gimbo wiki [home page](#)

From:  
<https://wiki.gimbo.org/> - **wiki.gimbo.org**

Permanent link:  
[https://wiki.gimbo.org/doku.php?id=public:texts\\_real](https://wiki.gimbo.org/doku.php?id=public:texts_real)

Last update: **13.03.2022 05:55**

