Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the universe trying to build bigger and better idiots. So far, the universe is winning.

Lisp isn’t a language, it’s a building material.

Walking on water and developing software from a specification are easy if both are frozen.

They don’t make bugs like Bunny anymore.

A C program is like a fast dance on a newly waxed dance floor by people carrying razors.

Computer science education cannot make anybody an expert programmer any more than studying brushes and pigment can make somebody an expert painter.

Computer system analysis is like child-rearing; you can do grievous damage, not because they expect to get paid

You can’t ensure success.

We are not shipping your machine!

We are not shipping your machine!

We are not shipping your machine!

We are not shipping your machine!

They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

If McDonalds were run like a software company, one out of every hundred Big Macs would give

sooner or later your nose will bleed.

I have always wished for my computer to be as easy to use as my telephone; my wish has come true because I can no longer figure out how to use my telephone.

I think Microsoft named.Net so it wouldn’t show up in a Unix directory listing.

Don’t worry if it doesn’t work right. If everything did, you’d be out of a job.

Mosher’s Law of Software Engineering

The use of OOBOL creates the mind; its teaching should therefore be regarded as a criminal offense.

E.W. Dijkstra

Talking about the correctness of a program is like discussing the correctness of a watch.

Talk is cheap. Show me the code.

The use of OOBOL creates the mind; its teaching should therefore be regarded as a criminal offense.

E.W. Dijkstra

The evolution of languages: FORTRAN is a non-typed language. C is a weakly typed language. Ada is a strongly typed language. C++ is a strongly typed language.

There are two ways of constructing a software design. One way is to make it so simple that there are obviously no deficiencies. The other way is to make it so complicated that there are no obvious deficiencies.

C.A.R. Hoare

There are two ways of constructing a software design. One way is to make it so simple that there are obviously no deficiencies. The other way is to make it so complicated that there are no obvious deficiencies.

C.A.R. Hoare

Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.

It’s a building material.

Most good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.

Linus Torvalds

The trouble with programmers is that you can never tell what a programmer is doing until it’s too late.

Seymour Cray

The trouble with programmers is that you can never tell what a programmer is doing until it’s too late.

Seymour Cray

The trouble with programmers is that you can never tell what a programmer is doing until it’s too late.

Seymour Cray

The trouble with programmers is that you can never tell what a programmer is doing until it’s too late.

Seymour Cray

To iterate is human, to recur divine.

L. Peter Deutsch

To iterate is human, to recur divine.

L. Peter Deutsch

To iterate is human, to recur divine.

L. Peter Deutsch

To iterate is human, to recur divine.

L. Peter Deutsch

To iterate is human, to recur divine.

L. Peter Deutsch

To iterate is human, to recur divine. A programming language should be like. But Java applications are good examples of what applications SHOULDN’T be like.

Programming is like kicking yourself in the face, sooner or later your nose will bleed.

It is easier to port a shell than a shell script.

I have only proved it correct, not tried it.

I invented the term ‘Object-Oriented’, and I can tell you I did not have C++ in mind.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.

The best programmers are not marginally better than merely good ones. They are an order-of-magnitude bigger, measured by whatever standard: conceptual creativity, speed, ingenuity of design, or problem-solving ability.

In practice, they’re not.