Computers, People and Programming

Chapter 1

Software

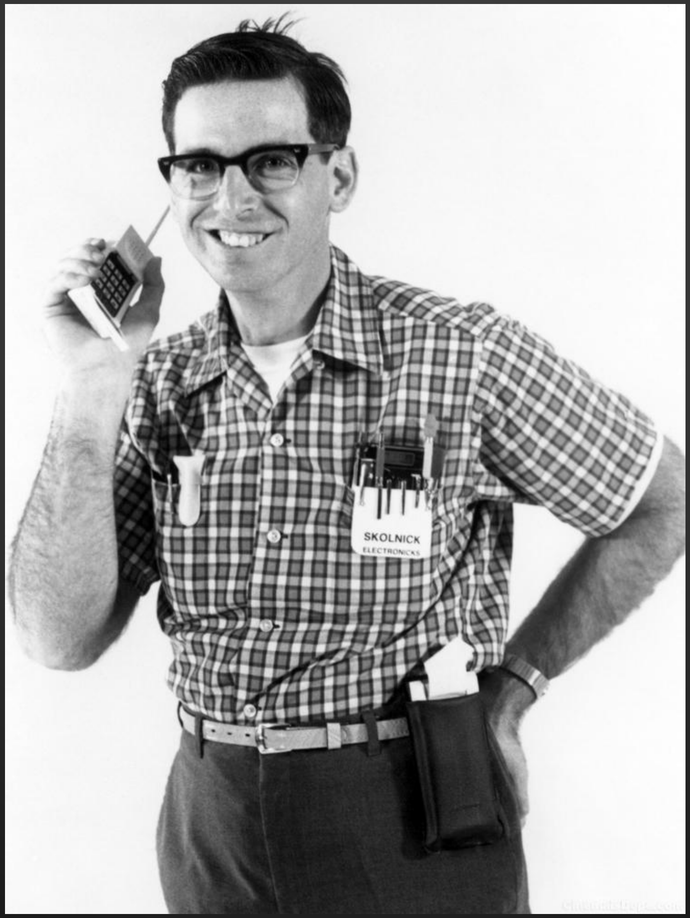
Software is a collection of programs running on some computers. p19

Computers are everywhere?

Where can you find computers?

Programmers

What does a programmer look like?



Programmers

What do real programmers look like?

Why do programmers like to program?

Most programmers don’t spend most of their time programming.

“That intellectual challenge is what many programmers refer to when they say that programming is interesting.” pg 23

A good programmer understands the role of code and programming technique in a project.

Computer Science

Programming is a tool.

Programming is where ideas and theories meet reality.

It must not degenerate into mere hacking; just get some code written, any old way that meets an immediate need.

Computers are everywhere!

Think of all the ways you interact with computers on a daily basis.

Is there anything you do that doesn’t use a computer?

Ideals for programmers

What do we want from our programs in general?

1. correctness
2. reliability
3. well designed
4. affordable
5. maintainable

Correctness and reliability

We want our program to do their jobs correctly and part of that is the program is also reliable.

A program that is correct, but only works on every other Tuesday is no good.

Well designed

A program that is correct and reliable, but doesn’t address a real need is useless.

If a program is too annoying to use, it becomes unused as well.

Affordable

A program that is too expensive to purchase also doesn’t get used.

Maintainable

A program that is too hard to maintain or too expensive to maintain becomes a useless program.

Only failed programs are not maintained or modified.

Why is this so hard?

“Computers are nit-picking, unforgiving, dumb beasts” pg 35.

We can’t simply tell the program to do some stuff and hope that it works.

We have to work out every detail and explain it very explicitly to the computer.

A program is a precise representation of our understanding of a topic.

Four stages of program development

Analysis
Design
Programming
Testing

Do this look familiar?

Analysis

What’s the problem?
What does the user want?
What does the user need?
What can the user afford?
What kind of reliability do we need?

Design

How do we solve the problem?

What should be the overall structure of the system?

Which parts does it consist of?

How do these parts communicate with each other?

How does the system communicate with its users?

Programming

Express the solution to the problem (the design) in code.

Write the code in a way that meets all the constraints (time, space, money, reliability, and so on).

Make sure that the code is correct and maintainable.

Testing

Make sure the system works correctly under all circumstances required systematically trying it out.

Programming plus testing is often called *implementation*.

One important note

These stages of development are not independent and do not occur strictly in sequence.

The important concept here is feedback.

We learn from experience and modify our behaviour based on what we learn.

Final thoughts

Programming is nothing more than an expression (in code) of some ideas.

Direct expression of ideas in code is a fundamental ideal of programming.

The ideal is that when we have an idea...then we want that something to exist in our program as a named entity (a type) providing the operations we think appropriate for it.

Last final thoughts

Programming is a part practical, part theory.

If you are just practical, you will produce non-scalable, unmaintainable hacks.

If you are just theoretical, you will produce unusable (or unaffordable) toys.