1. Introduction. Idealized computer models and abstract algorithms. The other speakers in this tutorial symposium will discuss recent theoretical developments in computer science. The studies they will describe have a combinatorial flavor which all mathematicians will find very familiar. My own task is the less tractable one of presenting the immediately pragmatic side of computer science; specifically, programming, and the content of investigations into programming technique. This is work, very directly rooted in the fertile muck of everyday industrial practice, out of which grow the more theoretical endeavors to be explained in the other lectures.

The goal at which programming technique aims is the rigorous, correct, and maximally clear and simple description of complex processes. This description must also be such as to permit efficient implementation on a computer. Rigor and correctness are necessary because programs are acted on by a device (the computer) lacking all but the most rudimentary ability to make inferences or to distinguish between the reasonable and the unreasonable. Clarity and simplicity of programs are essential because of the inherent limitations of the human mind, limitations which the activity of programming always makes painfully evident. It deserves to be noted that large programs, the most complex artificial objects known to mankind, generally threaten at every moment to submerge their creators in a flood of complications. Efficiency is important, not only because computer time is still an expensive commodity, but also because "effective" mathematical procedures can sometimes imply calculations so explosively large as to require astronomically long running times on any conceivable physical device.

In this expository talk I shall explain some of the notions which guide (or ought to guide) programmers in their attempts to develop...