PREFACE

When history passes judgment on the change-filled time in which we now live, one thing seems fairly certain—this will be called the "computer age."

Beginning around the middle of the twentieth century, and extending far enough into the future that few of us can imagine its ending, the computer age is the period of time during which computer power has dramatically changed, and continues to change, how the world functions. Few aspects of our lives are not touched by—and often, heavily influenced by—computers.

Was the computer's arrival a cosmic accident, an invention whose time simply came? Did the population explosion and the resulting social complexity make the computer a necessary tool to subdue that complexity inevitable? These questions are unanswerable, of course, but interesting to ponder.

The purpose of this book is not so much to address unanswerable questions as to capture concrete realities. The computer age, as any other, was made possible by the people who built it. The reality this book captures is that of the software pioneers of the field, the people who wrote the programs that made that complicated but very dumb product called a computer perform the magic that we have come to understand so well. Software, most would agree, makes the computer hardware world go 'round. But who are the pioneers who made the software world go 'round? And what were their experiences?

The time is right for such a book. In fact, the time is probably now or never. Given that the use of software to solve the world's problems began in the 1950s, those pioneers who began their professional careers at that time—perhaps then in their 20s—are now well into their 60s. Many of software's early innovators have already retired from the field, and some have died. Capturing the personal recollections of these pioneers now is essential, or they will be lost forever.

But is that enough of a reason for anyone to be interested in reading this book? Could not such a book simply be a compilation of dry-as-dust

information written by old fogies who are out of touch with today and can only vaguely recall those past days?

There are several important reasons why I believe this material is worth reading:

1. Professionals in our field are surprisingly misinformed about its history. As the editor of a professional journal, I frequently see papers cross my desk that describe computing's past as prologue to some new idea they are presenting—and they get that past totally wrong. It is time to record the truth of that prologue, if only to enable tomorrow's professionals to build more solidly on the realities of the past.
2. The old truism "everything old is new again" is just as true in computing as it is in other fields, especially in software. The stories in this book clearly show us that such modern concepts as data abstraction, modularity, and structured approaches date much earlier in the field than their appearance in the academic literature. In many fields of endeavor, practice often precedes theory (for example, the steam engine and human flight predated the theories of thermodynamics and aerodynamics), and the software field is no exception. These stories help to capture the field's true evolution.
3. Written history has a tendency to be cold and austere, full of too many dates and "major" events and too few human experiences and true turning points. In this book, we capture those human experiences and clues to those turning points through the personal recollections of those who lived them. This book is as alive as the people whose stories it contains.

There you have it. If you have read this far, perhaps your curiosity has at least been piqued by what we are trying to do here. So let me step aside and let you read on.

Welcome to the world of computing's past!

Robert L. Glass
At the end of the twentieth century