

Chapter 0. Preface

The goal of this text is to provide a rigorous introduction to computing, with an emphasis on computer programming that is both compelling and accessible to a range of potential computing students beyond the narrow constituency of traditional computer science students. It presents computing as an art, in the true classical sense, that provides an empowering form of personal computing. The language of instruction is Processing, a Java-based, visual programming language and development environment.

0.1. Computing as an Art

While computers have come to be used in nearly all fields of human endeavor, the study of computing, and in particular of computer programming, has continued to remain largely confined to the areas of business, technology, and science. This is unfortunate, for there really is no good reason why our notion of "computing" should remain limited to these areas.

Computing is an art, in the classical sense of the term. The English word "art" comes from the Latin word "ars," which referred not to our modern notion of a work of "fine art" but, rather, to a kind of practical knowledge. This sense of the term lives on in the continued designation of certain areas of study as "liberal arts," a phrase which originates from the belief that the acquiring of such knowledge is fundamental to the "liberty" — freedom, empowerment — of a human individual. We believe that computer programming is a liberal art in this sense, an area of practical and productive knowledge that can be liberating and empowering to individuals in a wide variety of arenas of human life, especially in a computer-saturated era such as ours.

The Latin word "ars" is, in turn, a translation of the Ancient Greek word "techné," which referred to the craft of putting theory (episteme) into practice. Techné was evidenced whenever someone exhibited the ability to achieve consistently good results in producing something, even though the particular details and circumstances entailed in this process vary. For example, the techné of making a flute was evidenced whenever such an artist was consistently able to design and produce good flutes, even when such particulars as the material, size, shape, and purpose of the flute would change. We believe that computing is a craft in this classical sense as well in that it entails a demonstrated knowledge of how to consistently produce computer programs that are of high quality, even though the particular details, purposes, and contexts of these programs are always changing.

0.2. "Personal" Computing?

If computing, and in particular programming, are both liberating and practical, why don't more people learn them?

With the rise of the commercial personal computing industry, beginning in the 1980s, the notion of "using" a computer became separated from "programming" the computer. Consumers were taught that programming was something they should not have to do — and would not want to do even if they knew how. Instead, they would purchase software applications that had been created by others. This was a practical approach but inevitably some users felt overly constrained by the restrictions built into their applications and found that they were unable to alter the software to fit certain aspects of their needs. Thus, with "personal computing," computer "users" became non-programmers and, as a result, came to be cut off from the real power of computing.

In contrast, during the decade prior to the emergence of this commercial computing in the early 1980s, a good number of visionaries in the field of computing had been operating according to a different notion of "personal computing." This included a great deal of remarkable work in pursuit of the goal of making computer programming simpler and more appealing to a broader range of persons. Though this work was largely abandoned with the rise of commercial personal computing, the goal of making programming a simpler and more widely appealing process continued to live on in the hearts of some. Indeed, the "Processing" programming language and environment succeeds in achieving exactly this: a more "personal" form of programming and, therefore, a more genuinely powerful form of personal computing.



Figure 0-1. A graphical hello-world program

0.3. About Processing

Processing was originally developed at the Massachusetts Institute of Technology for the purpose of making it easier for artists to create graphical art on a computer. It turns out, however, that Processing provides a wonderfully friendly and highly visual way for anyone to learn to program. Consider the simple program shown in Figure 0-1. This program, shown in the Processing development environment, implements a graphical version of the well-known “Hello, World!” program. In addition, the Processing environment provides support for developing animations. Processing has proven itself to be compelling and accessible to a wide range of computing students, beyond the traditional computer science student.

Because Processing is based on the powerful and widely-used "Java" programming language, Processing programs will run on any computer that supports Java. Further, students who have learned Processing will find it relatively easy to transition from Processing to the more powerful and general Java environment.