Table of Contents

How to be a Programmer: Community Version

Robert L. Read with Community

Introduction

To be a good programmer is difficult and noble. The hardest part of making real a collective vision of a software project is dealing with one's coworkers and customers. Writing computer programs is important and takes great intelligence and skill. But it is really child's play compared to everything else that a good programmer must do to make a software system that succeeds for both the customer and myriad colleagues for whom she is partially responsible. In this essay I attempt to summarize as concisely as possible those things that I wish someone had explained to me when I was twenty-one.

This is very subjective and, therefore, this essay is doomed to be personal and somewhat opinionated. I confine myself to problems that a programmer is very likely to have to face in her work. Many of these problems and their solutions are so general to the human condition that I will probably seem preachy. I hope in spite of this that this essay will be useful.

Computer programming is taught in courses. The excellent books: The Pragmatic Programmer [Prag99], Code Complete [CodeC93], Rapid Development [RDev96], and Extreme Programming Explained [XP99] all teach computer programming and the larger issues of being a good programmer. The essays of Paul Graham [PGSite] and Eric Raymond [Hacker] should certainly be read before or along with this article. This essay differs from those excellent works by emphasizing social problems and comprehensively summarizing the entire set of necessary skills as I see them.

In this essay the term boss is used to refer to whomever gives you projects to do. I use the words business, company, and tribe, synonymously except that business connotes moneymaking, company connotes the modern workplace and tribe is generally the people you share loyalty with.

Welcome to the tribe.

Contents