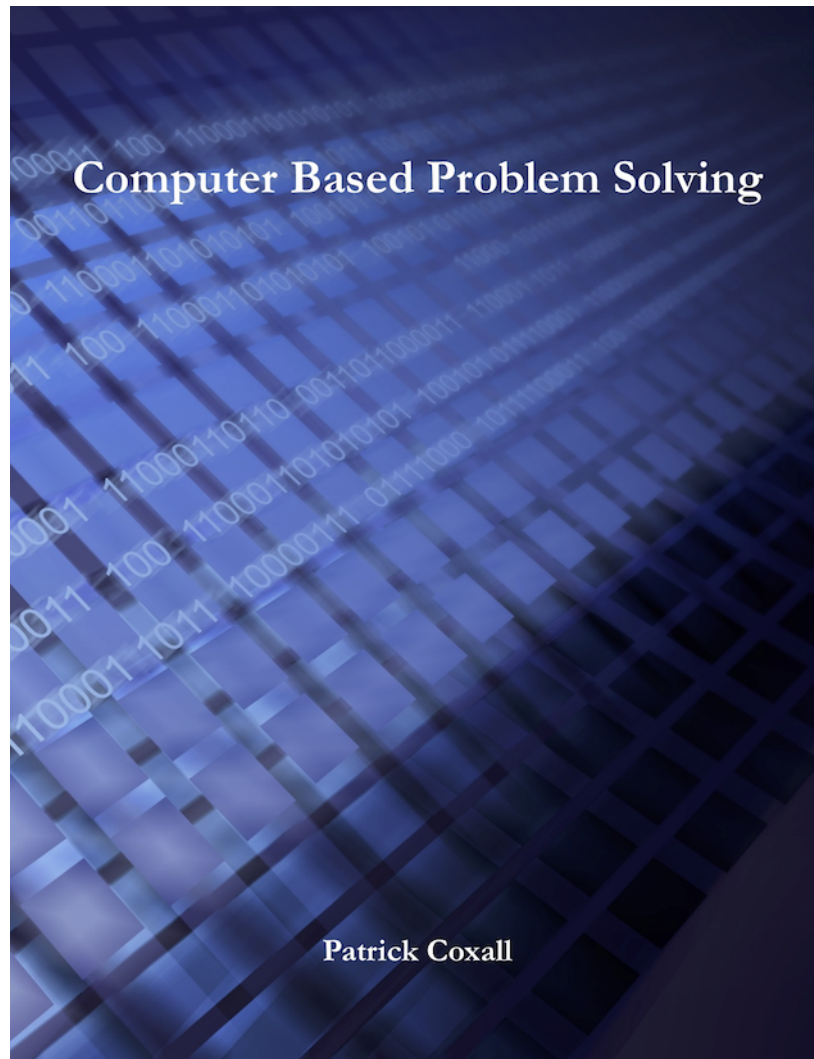

Computer Based Problem Solving

Patrick Coxall

Jul 28, 2020

Contents

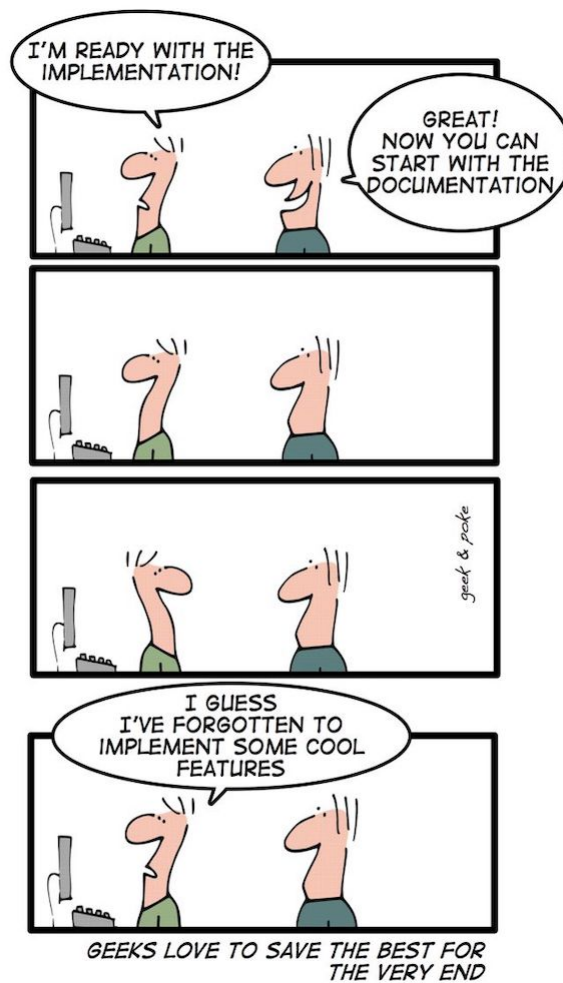
1	Preface	3
2	Introduction	5
2.1	What is programming	7
2.2	Goal of this book	8



The goal of this book is to take students from the point of never having done any formal programming and lead them first through a structured method of problem solving (Input-Process-Output and Top-Down design), then into basic Structured Programming and then into the early basics of Object Oriented Programming (or OOP). If this book is used to teach a high school course in computer programming, there are likely many other learning outcomes that students are required to do that are not presented in this book. The focus of this book is strictly on solving problems with computer programming.

CHAPTER 1

Preface



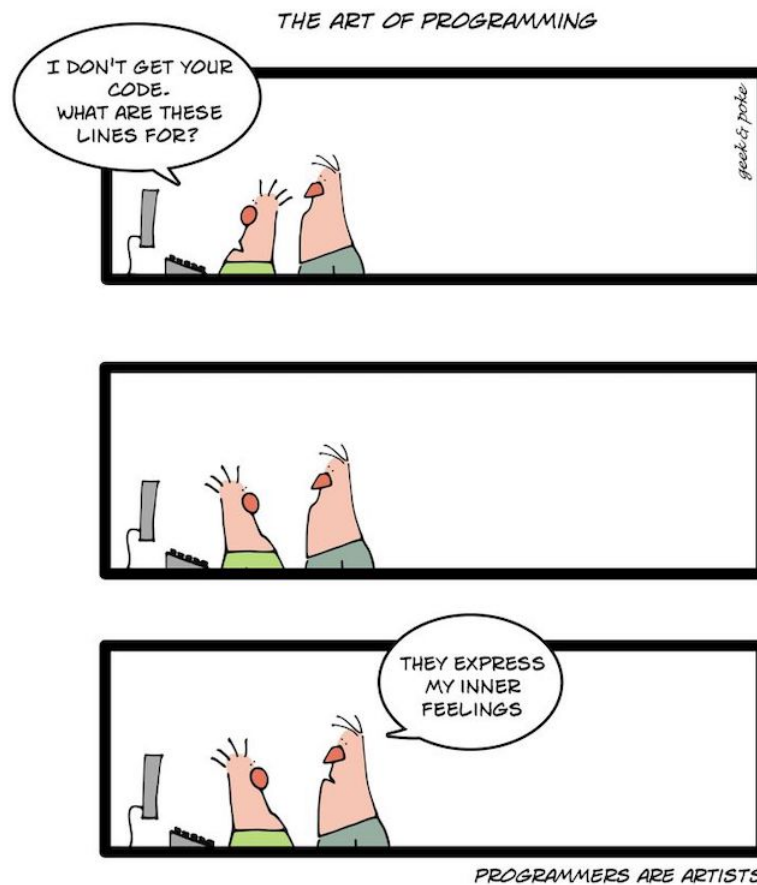
It should be remembered that the focus of this book is to teach students how to program, not to just teach them a programming language. To do this the focus is on “Problem Solving”, using a computer program as a problem solving aid. Programming languages change over time and come and go but a good foundation of programming concepts and how to solve a problem will allow anyone to get over the syntax of a new programming language.

This book does not include any instructions on how to load, use, create GUIs or any other housekeeping of any particular language. There are many other resources that can aid both students and teachers alike for this.

Within the text you will see words or groups of words that are hyperlinked to [Wikipedia](#). The point of linking to Wikipedia is to give additional information about a topic if the reader is unsure about the concept. Please note that I do not have control over what is placed on Wikipedia and although it seemed useful and correct when I looked at the link, these pages are changing all the time. Despite this, the information is usually correct and can be very helpful.

CHAPTER 2

Introduction

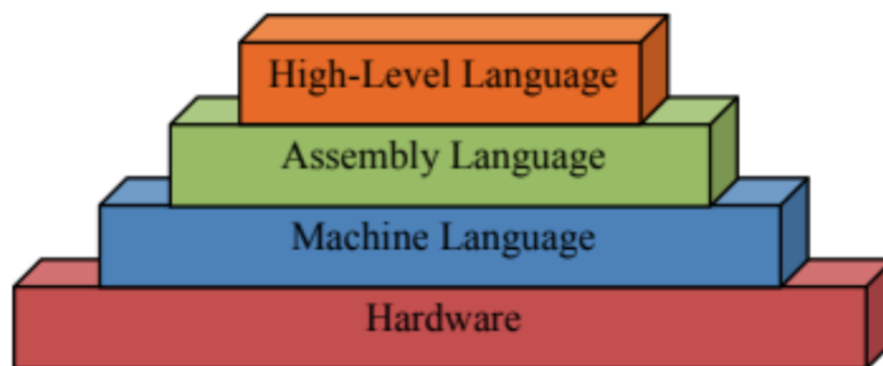


Problems have been around for as long as people have been around. The process of solving a problem is not something new. Using a computer to aid in solving a problem is new. Modern electronic computers have only been around since the Second World War (1939-1945), which might seem like a long time ago to you but in the history of the human race it is a very short time. The purpose of this book is to help you learn to structure your problem solving method, so that you can consistently develop a verifiable solution that will solve a problem and in the process, use the computer to help you more easily and quickly solve that problem.

2.1 What is programming

Before you can actually start to write programs on a computer to help you solve problems, it would be nice to know what programming really is! We all use and see computers every day and hear people say how smart computers are. Actually, computers are not very smart at all! A computer, broken down into its most basic form is nothing more than a bunch of tiny electronic switches, called **transistors**, that can be set to either a 1 or 0 (on or off, also known as **binary**). By getting the computer to set these tiny switches on or off in a certain pattern, you can get the computer to actually do something useful, like show a picture on the screen that you have taken. The computer does not know how to do this by itself though.

To communicate with your friends, one way for them to understand what you mean is for you to talk to them. To keep things simple, you both usually talk in the same language. Since a computer is just a bunch of switches, it does not understand English, so you have to talk to it in a language that it does understand. Computers use a language called **machine language**, made up of just the 1's and 0's mentioned above. Trying to talk in machine language is quite difficult, easy to make mistakes in and tedious. People quickly realized they needed a better way. Assemble languages were created that used very simple instructions (like DCR C ; which decreases C counter by one). This assemble language is then run through a program that converts it to machine code. Eventually people wanted an even easier to understand language. To help people talk to a computer a **high-level programming language** is normally used, that is then translated into machine language so that the computer can understand what to do. This high-level language comes in many different variations and is normally just called a **programming language** and you have probably already heard of some of them (Java, C++, Python, ...). Just like there are many different languages that people speak around the world (English, French, Spanish, ...), there have been many different programming languages developed to help people instruct computers in what to do. The purpose of a programming language is to make it easier for a human to tell a computer what to do, by not having to talk machine language.



A person that uses a programming language to instruct a computer what to do is called a **programmer**. The programmer solves whatever problem they are working on, then writes the instructions that the computer is to follow in the programming language that they have chosen. Then the computer translates the instructions into machine language (the language that the computer actually understands) and the computer performs these actions.

2.2 Goal of this book

The goal of this book is to make you a “good” programmer. Despite the fact that a normal high school semester courses run for about 90 days, you will not become an expert programmer in just one semester. It has been said that it takes around 10,000 hours to really become proficient at anything and programming is no different. By the end of this book you will be very much on your way and have a good foundation in the skills you will need. The important thing to remember is that the point is not to teach you a specific programming language, since programming languages come and go and change over time. This is just like real languages.

Although it is the official language of the Catholic Church, not too many people go around on the streets and speak Latin to their friends. Many years ago, it might have been common but not today. The tools you will learn from this book are good programming techniques. These tools will be useful no matter what programming language you are using. Just like in the real world, you cannot be called a “linguist” if you only know one language. The same thing is true for a programmer; knowing more than one language is essential. The fortunate thing is that if you know how to program and know one programming language, picking up a second one is much easier. The cornerstone of being a good programmer is to be able to solve problems in a logical and systematic way and hopefully have fun in the process.