

A GUIDE TO PROGRAMMING AND CONCEPTS



VIVEK PARIHAR

Copyright © 2019, Vivek Parihar
All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording or any information storage and retrieval system now known or to be invented, without permission in writing from the publisher, except by a reviewer who wishes to quote brief passages in connection with a review written for inclusion in a magazine, newspaper or broadcast.

Published in India by Prowess Publishing,
YRK Towers, Thadikara Swamy Koil St, Alandur,
Chennai, Tamil Nadu 600016

ISBN: 978-93-89097-80-1
eISBN: 978-93-89097-81-8

Library of Congress Cataloging in Publication

Contents

Preface

Chapter 1: Programming

- 1.1 What is Programming?
- 1.2 What is a Computer Program?
- 1.3 Analogy for Programming
- 1.4 The Natural Language of the Computer?
- 1.5 What is Programming Language?

Chapter 2: Some Terminologies

- 2.1 Computer
- 2.2 Program
- 2.3 Programming Languages
 - 2.3.1 Low-level Language
 - 2.3.2 High-level Languages
- 2.4 What makes a good programming language?
 - 2.4.1 Clarity, simplicity and unity
 - 2.4.2 Orthogonality
 - 2.4.3 Naturalness of application
 - 2.4.4 Support for abstraction
 - 2.4.5 Ease of program debugging
 - 2.4.6 Programming environment
 - 2.4.7 Portability of program

2.4.8 Cost of using

2.5 Converting to Machine Language

2.5.1 Compiler

2.5.2 Interpreter

Chapter 3: Reasons for Studying Concepts of Programming Languages

3.1 Increased ability to express ideas

3.2 Improved background for choosing appropriate languages

3.3 Greater ability to learn new languages

3.4 Understand significance of Implementation

3.5 Ability to design new languages

Chapter 4: Programming Domains

Chapter 5: Language Evaluation Criteria

Criteria 1: Readability

Criteria 2: Writability

Criteria 3: Reliability

Criteria 4: Cost

Chapter 6: How do Programming Languages Differ

6.1 Common and Uncommon Constructs

6.2 Language Categories/Paradigms

6.2.1 Imperative

a. Procedural Programming

b. Object Oriented Programming (OOP)

6.2.2 Declarative

- a. Functional Programming
 - b. Logic Programming
- 6.2.3 Concurrent
- 6.2.4 Scripting
- 6.3 Values and Types**
- 6.4 What is a Data Type?**
 - 6.4.1 Int
 - 6.4.2 Strings
 - 6.4.3 Boolean
 - 6.4.4 List
 - 6.4.5 Trees
- 6.5 What is a Type System?**
 - 6.5.1 Static
 - 6.5.2 Dynamics
 - 6.5.3 Weak
 - 6.5.4 Strong
 - 6.5.5 Persistent
 - 6.5.6 Transient
- 6.6 Variables & Storage**
 - 6.6.1 Creating variables
 - 6.6.2 Assigning Value to Variables
 - 6.6.3 Key points about variables
 - 6.6.4 Access stored values in variables
 - 6.6.5 Variables in Java
 - 6.6.6 Variables in Python

Chapter 7: Language Design Trade-offs

- 7.1 Procedural programming
- 7.2 Functional programming
- 7.3 Declarative programming
- 7.4 Classic Object-Oriented programming
- 7.5 Prototypical OO programming
- 7.6 Aspect-oriented programming
- 7.7 Some differences between multiple criterias
 - Reliability vs. cost of execution
 - Readability vs. writability
 - Writability (flexibility) vs. reliability

Chapter 8: Implementation Methods

- 8.1 Compilation
- 8.2 Pure Interpretation
- 8.3 Hybrid Implementation Systems

Chapter 9: Programming Environments

- 9.1 Features of IDEs

Chapter 10: Programming Language History

The genealogy tree of programming languages

10.1 Generations of Programming Languages

10.1.1 First Generation Language (Machine language) First Generation Language (Machine language) First Generation Language (Machine language) First Generation Language (Machine language)

10.1.2 Second Generation language (Assembly Language)

- 10.1.3 Third Generation languages (High-Level Languages)
- 10.1.4 Fourth generation language (Very High-level Languages)
- 10.1.5 Fifth generation language (Artificial Intelligence Language)

Chapter 11: Career Scope of Language

11.1 What language do you learn first?

11.2 How to prepare yourself for your first line of code?

Step #1: Installing the Programs

Step #2: Deconstruct the Skill

Step #3: Learn the Skills You Need

Step #4: Join the Programming Community

Now take action!

11.3 What basic tools do you need to start programming?

1. Git and Github
2. SQL
3. Python
4. Text Editors (SubLime)
5. Linux commands
6. SVN
7. Powershell or Cygwin Or Bash (CLI)

References



Programming

1.1 What is Programming?

“Programming is the act of instructing computers to carry out tasks.” It is often referred to as coding and is the process of taking an algorithm and encoding it into a notation. Without an algorithm there can be no program.

Computer science is not the study of programming. Programming, however, is an important part of what a computer scientist does. Programming is often the way that we create a representation for our solutions. Therefore, this language representation and the process of creating it becomes a fundamental part of the discipline.

1.2 What is a Computer Program?

A computer program is a sequence of instructions that the computer executes. Most computer devices require programs to function properly.

A collection of computer programs, libraries, and related data are referred to as software. Computer programs may be categorized along functional lines, such as application software and system software. The underlying method used for some calculation or manipulation is known as an algorithm.

1.3 Analogy for Programming

There are patterns to our everyday lives. The universe operates in a somewhat predictable way; For example—day and night, seasons, sunrise and sunset. People go through routines such as rising in the morning, going to school or to work. We get instructions from other people such as our superiors at work. How we cook certain recipes can be explained in finite steps.

Second, every time we use smart devices, some code is running in the background. Moving a mouse pointer from one part of your computer screen to the other may seem like a simple task, but in reality, so many lines of code just ran. An act as simple as typing letters into Google Docs leads to lines of code being executed in the background. It's all code everywhere.

1.4 The Natural Language of the Computer?

Machines have their natural language like humans do. Computers do not understand the human language. The natural language of computers is the binary code—1 and 0. These represent two states: on (1) and off (0).

1.5 What is Programming Language?

The Programming Language is an Artificial Language that can be used to instruct a Computer to perform a Particular task. To be considered general Programming Language, it must be computationally complete, or Turing-Complete.

It is nevertheless common to regard some Language that are not computationally complete, like database query language and other domain-specific languages as programming language as well. Just as English is the international language of business and French is the language of love, different programming Language is better suited for different purposes.

You've Just Finished your Free Sample

Enjoyed the preview?

Buy: <http://www.ebooks2go.com>