
FRONT COVER:

Terms and Conditions

LEGAL NOTICE

The Publisher has strived to be as accurate and complete as possible in the creation of this report, notwithstanding the fact that he does not warrant or represent at any time that the contents within are accurate due to the rapidly changing nature of the Internet.

While all attempts have been made to verify information provided in this publication, the Publisher assumes no responsibility for errors, omissions, or contrary interpretation of the subject matter herein. Any perceived slights of specific persons, peoples, or organizations are unintentional.

In practical advice books, like anything else in life, there are no guarantees of income made. Readers are cautioned to rely on their own judgment about their individual circumstances to act accordingly.

This book is not intended for use as a source of legal, business, accounting or financial advice. All readers are advised to seek services of competent professionals in legal, business, accounting and finance fields.

You are encouraged to print this book for easy reading.

Table Of Contents

Foreword

Chapter 1:

Understanding Programming Concepts And How They Work

Chapter 2:

Techniques Of Writing A Program

Chapter 3:

The List Of Programming Languages

Chapter 4:

How To Choose The Right Compiler

Chapter 5:

What Is An Interpreter

Chapter 6:

Writing Your Program With An Editor

Chapter 7:

The Functions Of A Debugger

Chapter 8:

Ease Your Burden With Components

Chapter 9:

Optimizing Your Program With Profiler

Chapter 10:

Installing Your Program

Wrapping Up

Foreword

There are several different concepts that an individual needs to understand before being able to tackle the issue of programming concepts and how they unfold.



Programming For A Freshman

A Beginner's Guide To Programming

Chapter 1:

Understanding Programming Concepts And How They Work

Synopsis

Programming concepts may include information on how to write Web Logic startup scripts, creating Web services within the Web Logic site, steps on how to understand how the particular programs being written works and many more platforms that require thorough understanding.



Where To Start

Some would say that computers are actually very dumb tools that cannot function without a preset guidance module incorporated into its system. This would literally mean that a computer program will not be able to be of any use without actually being “told” what to do. A computer is unable to act or think for itself, unless the appropriate accompanying software used clearly depicts step by step what needs to be done for any particular activity. Therefore, to understand how the programming works, the user will first have to understand the concepts required to make the system work in the first place.

Some of these should ideally include understanding the terms and programs. Among which would be the famous and basic word processor program, where the input of characters typed using the keyboard will execute the formats of the text, correct any mistakes in spelling and grammar and create the output of displays and points that are neatly organized in text form to be readable for the user.

There are also the game programs where the keystrokes or joystick movements are part of the input which constitutes the calculation of speed and distance the animations take on the screen. These calculations have the output of actions and movements of the particular animation on the screen being viewed.

The stock market predictor is another program that has great input capacity and the ability to give information on the current and past

prices of stocks. This is where the programs tries to recognize the trend in the price fluctuation and has the capability of predicting the future probable price of stocks.



Chapter 2:

Techniques Of Writing A Program

Synopsis

There are several important aspects to consider before actually commencing on writing a particular program. Without these considerations in place, the individual could end up writing a program that will be rendered useless or inadequate in its ability to perform its required actions.



Helpful Techniques

Some of the steps taken first should be to identify the problem so that the program can be designed to tackle the basic needs of the user who chooses to use the programs. Then, there would also be a need to identify the user in terms of their reasons for actually using the program in question. There will also be a need to identify the computer that is being used with the program. Computers such as Windows, Macintosh, mainframe computers, and computers running Linux, Handheld Palm or Pocket PC or any other super computers should also be taken into account. This is due to the fact that certain programs are not compatible with some of the tools the users are currently using. The programming skill of the individual should also be considered, as the competency level of the eventual outcome of the programs should be able to address all the various issues that might arise during the use of the particular program.

Understating that a program is only useful if it is able to address the issues it is being used for is the ultimate goal for its existence. If the program is not able to address this in its completed form, then the programs is usually deemed unfit for use. Therefore, it is important to ensure the writing of the programs covers every detail and reason for the program's usage before it is released for use.

Chapter 3:

The List Of Programming Languages

Synopsis

As with the multitude of human languages used for communication purposes, the computer systems also requires and uses various languages to communicate its information to the user. These languages are usually based on certain syntactic and semantic rules, which contribute to the defining of the meaning of each programming language and its individual implementation.



How Its Made

The programming languages are put in place to provide the computer programmers with the means to create and express computer algorithms.

The following are some of the more popularly used languages:

APL – this array of programming languages works simultaneously on a multiple array of data platforms and is interpretive, interactive and functional as a programming language.

AutoIt – this freeware automation language is for Microsoft Windows and its main intent is to create automation scripts that can be used for the execution of some repetitive tasks that are within the Windows application.

Basic – also otherwise referred to as the beginner’s all-purpose symbolic instruction code, this was designed to provide non scientifically inclined users access to computers. Microsoft Visual Basics is predominantly using this program.

Eiffel – this is an object orientated programming language that is ISO standardized and developed to be able to provide extensible and reusable software. It is popularly used in industries such as finance, aerospace and video gaming. It has since evolved to include many other functional programming features such as Ubercode, which is a high level platform sharing language.

Forth – this structured imperative programming language is based on the implementation idea of stacks, while supporting an interactive execution of commands, as well as a compilation of sequences of commands at each use.

Frink – this program is based on the Java Virtual Machine and focuses mainly on science and engineering elements. Its main feature of tracking unit measurable through calculations enables the quantities to contain their respective unit's within the measurements keyed.

This preview copy ends here and to get the full product, please go to
(the website URL).