**Absolute Beginner’s Guide to Python Programming**

**Rev. 0 – 7/17/24 – Created by Paste from One Note**

Paste:

To Do Dedication

To Do Preface -- Why You Want To Read This Book

To Do Table of Contents

To Do Words of Wisdom

To Do The Art of Falling:  A Programmer’s Journey to Mastery

To Do **Introduction**

To Do Why Python?

To Do What Makes Python Special?

To Do Discovering Python's Applications

To Do Goals of This Book

To Do How to Use This Book

To Do Where Can I Find the Source Code for This Book?

To Do **Chapter 1: The First Steps**

To Do 1.1 Understanding the Python Interpreter

To Do 1.2 Choosing an IDE

To Do 1.3 Getting Started with Visual Studio Code

To Do 1.4 Windows: Installing Python and VS Code

To Do *~~1.5 macOS: Installing Python and VS Code~~*

To Do *~~1.6 Linux: Installing Python and VS Code~~*

To Do 1.7 Configuring Visual Studio Code for Python Development

To Do 1.8 The Classic "Hello, World!"

To Do 1.9 Executing a Script

To Do 1.10 Taking User Input

To Do 1.11 Displaying Output

To Do 1.12 Chapter Summary

To Do **Chapter 2: The Foundations of Python**

To Do 2.1 Variables and Data Types: The Building Blocks

To Do 2.2 String Formatting in Python: F-Strings and format() Method

To Do 2.3 Understanding Immutability

To Do 2.4 Comments: Your Code’s Annotation

To Do 2.5 Control Structures: Making Decisions in Your Code

To Do 2.6 Understanding Python Syntax: Indentation and the Colon

To Do 2.7 if, elif, and else

To Do 2.8 Calculations in Python

To Do 2.9 Python Operators

To Do 2.10 Ternary Conditional Expression

To Do 2.11 Chapter Summary

To Do 2.12 Suggested Exercises

To Do 2.13 Project: Loan Payment Calculator

To Do 2.14 References

To Do **Chapter 3: Exploring Python with Loops and Collections**

To Do 3.1 Lists

To Do 3.2 Understanding Multi-dimensional Arrays in Python

To Do 3.3 Introduction to Tuples

To Do 3.4 Dictionaries: Mapping Keys to Values

To Do 3.5 Comparing Dictionaries to Lists and Tuples

To Do 3.6 Sets: Unleashing the Potential of Unique Collections

To Do 3.7 Performance Considerations

To Do 3.8 Summarizing Collections

To Do 3.9 Loops: The Power of Repetition

To Do 3.10 While Loops

To Do 3.11 Loop Control Statements

To Do 3.12 Nesting Loops

To Do 3.13 Chapter Summary

To Do 3.14 Suggested Exercises

To Do 3.15 Project: Build a Python Quiz Game

To Do **Chapter 4: The Power of Functions, Modules, Packages and Lambdas**

To Do 4.1 Functions

To Do 4.2 Scope and Lifetime of Variables

To Do 4.3 Modules

To Do 4.4 Packages

To Do 4.5 Lambda Functions

To Do 4.6 Enhancing Lambda Functions with filter, map, and sorted

To Do 4.7 Chapter Summary

To Do 4.8 Suggested Exercises

To Do 4.9 Project: Tic-Tac-Toe Game

To Do 4.10 Game Design and Flow

To Do 4.11 Step-by-Step Guide to Building the Game

To Do 4.12 Introducing an AI Opponent

To Do 4.13 Step-by-Step Guide to Building the AI

To Do 4.14 Explore Advanced AI Techniques

To Do **Chapter 5: Data Storage: Text & JSON Files**

To Do 5.1 Basic File Operations

To Do 5.2 Handling Errors with try/except/finally

To Do 5.3 Working with Text Files

To Do 5.4 Working with JSON Files

To Do 5.5 Other Types of Structured Files

To Do 5.6 Chapter Summary

To Do 5.7 Chapter Exercises

To Do 5.8 Chapter Project: Personal Expense Tracker

To Do **Chapter 6: Navigating Object-Oriented Programming**

To Do 6.1 What are Classes and Objects?

To Do 6.2 Define a Class

To Do 6.3 Creating and Using Objects

To Do 6.4 Understanding Composition: Classes Within Classes

To Do 6.5 Wrapping Up Encapsulation

To Do 6.6 Embracing Inheritance

To Do 6.7 Wrapping up Inheritance

To Do 6.8 Exploring Polymorphism

To Do 6.9 Wrapping up Polymorphism

To Do 6.10 Mastering Abstraction

To Do 6.11 Abstract Classes vs Interfaces

To Do 6.12 Wrapping up Abstraction

To Do 6.13 Chapter Summary

To Do 6.14 Chapter Exercises

To Do 6.15 Chapter Project: Enhanced Expense Tracker with OOP

To Do 6.16 Step-by-Step Guide:

To Do  **Chapter 7: Graphical User Interfaces with Tkinter**

To Do 7.1 Verifying Tkinter’s Availability

To Do 7.2 Troubleshooting Common Issues

To Do 7.3 Your First Tkinter Application

To Do 7.4 Layout Management

To Do 7.5 Choosing the Right Layout Manager

To Do 7.6 Widgets: The Building Blocks of GUIs

To Do 7.7 Message Boxes

To Do 7.8 Event Handling in Tkinter

To Do 7.9 Event Sequences and Modifiers

To Do 7.10 Advanced Widget Customization and Styles

To Do 7.11 Using ttk (Themed Tk) for a Modern Look

To Do 7.12 Dynamic Widget Updates

To Do 7.13 Integrating External Data and Services in Tkinter Applications

To Do 7.14 Deploying Tkinter Applications: Best Practices and Sharing with Users

To Do 7.15 Chapter Summary

To Do 7.16 Suggested Exercises

To Do 7.17 Chapter Project – Convert Tic-Tac-Toe to Tkinter

To Do 7.18 Designing the Tic-Tac-Toe GUI

To Do 7.19 Review of Console Application

To Do 7.20 Updating the Main File - Initializing the Tkinter Loop

To Do 7.21 Step-by-Step Instructions

To Do **Chapter 8: Exploring the Future: Where to Go from Here**

To Do 8.1 Fields Where Python is Used

To Do 8.2 Web Development

To Do 8.3 Data Science

To Do 8.4 Artificial Intelligence (AI)

To Do 8.5 Scientific Computing

To Do 8.6 Automation and Scripting

To Do 8.7 Software Development

To Do 8.8 Game Development

To Do 8.9 Network Programming

To Do 8.10 Finance

To Do 8.11 Chapter Summary

To Do **Conclusion**

To Do Embrace the Endless Possibilities

To Do The Art of Falling

To Do Building a Strong Foundation

To Do The Balance of Learning

To Do Encouragement for the Future

To Do Final Thoughts

To Do **Appendix A: Exercise Solutions**

To Do Chapter 2 Solutions

To Do Chapter 3 Solutions

To Do Chapter 4 Solutions

To Do Chapter 5 Solutions

To Do Chapter 6 Solutions

To Do Chapter 7 Solutions

To Do **Appendix B: GitHub Primer for Downloading Source Code**

To Do Accessing and Downloading Code from GitHub

To Do Tips for Using Downloaded Code

To Do No GitHub Account Required

To Do **Appendix C: Self-Documenting Code:**

To Do Writing Code That Explains Itself

To Do Key Characteristics of Self-Documenting Code

To Do Benefits of Self-Documenting Code

To Do **Appendix D: Choosing the Right IDE**

To Do Visual Studio Code (VS Code)

To Do PyCharm Jupyter

To Do Notebook Spyder

To Do Thonny IDLE (Python's Built-in IDE)

To Do PyDev (for Eclipse)

To Do Anaconda Navigator

**Glossary**