Part 1: Beginner Exercises

Strings & Printing

1. **Hello, World!:** Print the string "Hello, World!".

User Input
2) User Input and Greeting: Ask the user for their name and print a personalized greeting.
Numbers, Operators, & Conditionals
3) Simple Calculator: Create a program that takes two numbers as input and performs addition, subtraction, multiplication, and division.
Conditionals & Modulo Operator
4) Even or Odd: Check if a given number is even or odd.
String Slicing
5) String Reversal: Write a function that reverses a given string.
Mathematical Operations & Variables
6) Area of a Circle: Calculate the area of a circle given its radius.
Functions & Mathematical Formulas
7) Temperature Converter: Convert a temperature from Celsius to Fahrenheit and vice versa.
Conditional Logic
8) Largest of Three Numbers: Find the largest number among three user-provided numbers.
Strings & Loops
9) Vowel Counter: Count the number of vowels in a given string.
Loops
10) Factorial Calculator: Calculate the factorial of a number using a loop.
Loops & Sequences
11) Fibonacci Sequence: Generate the first 'n' numbers of the Fibonacci sequence.
String Manipulation
12) Palindrome Checker: Check if a string is a palindrome (reads the same forwards and backward).
Lists & Loops
13) List Sum: Calculate the sum of all elements in a list.
Lists, Loops, & Conditionals
14) Find Maximum in List: Find the largest element in a list without using the built-in max() function.
Lists & Counting
15) Count List Elements: Count the occurrences of a specific element in a list.
Lists & Sets
16) Unique Elements: Create a new list containing only the unique elements from an original list.
Dictionaries
17) Dictionary Basics: Create a dictionary representing a person (name, age, city) and print its values.
String Methods

18) Word Count: Count the number of words in a sentence.
Mathematical Calculations
19) Simple Interest Calculator: Calculate simple interest given principal, rate, and time.
Random Module & While Loops
20) Guess the Number Game: Generate a random number and have the user guess it, providing "higher" or "lower" hints.
Loops & String Formatting
21) Multiplication Table: Print the multiplication table for a given number.
Functions, Loops, & Logic
22) Check for Prime Number: Write a function to check if a number is prime.
String Indexing & Slicing
23) String Slicing: Given a string, print the first 3 characters and the last 3 characters.
List Comprehensions
24) List Comprehension: Use list comprehension to create a list of squares of numbers from 1 to 10.
Advanced Loops & Conditionals
25) FizzBuzz: Print numbers from 1 to 100. For multiples of 3, print "Fizz". For multiples of 5, print "Buzz". For multiples of both, print "FizzBuzz".

Part 2: Intermediate Exercises

File I/O (Reading & Writing)
26) Read and Write to File: Write a program to read content from a text file and write it to a new file.
File I/O & String Splitting
27) Count Lines in File: Count the number of lines, words, and characters in a text file.
CSV Module
28) CSV File Reader: Read data from a CSV file and print it in a formatted way.
JSON Module
29) JSON Data Handling: Read data from a JSON file, parse it, and access specific values.
Object-Oriented Programming (Classes)
30) Basic Class: Create a Car class with attributes like make, model, and year, and methods like start_engine().
Object-Oriented Programming (Inheritance)
31) Inheritance: Create an ElectricCar class that inherits from the Car class and adds a battery_size attribute.
Error & Exception Handling
32) Exception Handling: Modify the Simple Calculator (exercise 3) to handle ZeroDivisionError and ValueError.
Dictionaries & File I/O
33) Contact Book: Create a simple command-line contact book application using a dictionary to store contacts. Save and load contacts from a file.
Lists & File I/O
34) To-Do List Application: A command-line to-do list where users can add, view, and delete

tasks. Persist tasks to a file.

Algorithms & String Manipulation

35) Caesar Cipher: Implement the Caesar cipher for basic string encryption and decryption.

Random Module & String Constants

36) Password Generator: Create a program that generates a random password of a specified length, including a mix of letters, numbers, and symbols.

APIs & requests Library

37) Weather API Client: Use the requests library to fetch weather data for a given city from a free weather API.

Web Scraping (requests & BeautifulSoup)

38) Web Scraper: Use requests and BeautifulSoup to scrape and print all the headlines from a news website's homepage.

Lambda Functions

39) Lambda Functions: Rewrite a simple function (e.g., one that doubles a number) using a lambda function.

Functional Programming

40) Map, Filter, Reduce: Use map to square all numbers in a list, filter to get all even numbers, and functools.reduce to find the product of all numbers.

Recursion

41) Recursive Factorial: Re-implement the factorial calculator (exercise 10) using recursion.

Random Module

42) Dice Rolling Simulator: Simulate rolling two dice and display the results.

Game Logic & Loops

43) Hangman Game: Create a command-line Hangman game.

Game Logic & 2D Lists

44) Tic-Tac-Toe Game: Implement a two-player Tic-Tac-Toe game on the command line.

OS Module

45) Directory Tree Lister: Write a script that lists all files and subdirectories within a given directory.

requests Library & Binary File I/O

46) Image Downloader: Write a script that downloads an image from a URL and saves it locally.

Time Module

47) Simple Stopwatch: Create a command-line stopwatch to measure elapsed time.

State Management & Dictionaries

48) Text-Based Adventure Game: Create a simple adventure game where the user makes choices to navigate through a story.

APIs & requests Library

49) URL Shortener: Connect to a URL shortener API (like tinyurl) to create a short URL for a given long URL.

Regular Expressions (RegEx)

50) Regular Expression Validator: Write a function that uses regex to validate if a string is a valid email address.

Part 3: Project Ideas

Once you're comfortable with the exercises, you can tackle these larger projects to build your portfolio.

AI-Focused Projects

- **Sentiment Analysis Tool**: A script that takes a block of text (e.g., a product review) and classifies its sentiment as positive, negative, or neutral using a library like NLTK or TextBlob.
- **Simple Chatbot**: Create a rule-based chatbot that can answer predefined questions. For a more advanced version, integrate with a platform like Dialogflow.
- **Image Classifier with a Pre-trained Model**: Use a library like TensorFlow or PyTorch to load a pre-trained model (e.g., MobileNet) and build an application that can classify images you provide.

Cybersecurity-Focused Projects

- **Network Port Scanner**: A tool that scans a target IP address to identify open ports. This is a fundamental tool in network reconnaissance.
- **File Encryption/Decryption Tool**: An application that can encrypt a file using an algorithm like AES and a user-provided password, and then decrypt it back.
- **Password Strength Checker**: A tool that analyzes a password to determine its strength based on criteria like length, character types, and common patterns.
- **Simple Keylogger**: (For educational purposes only and to be used on your own machine). A script that records keystrokes and saves them to a log file to help understand malware behavior and system monitoring.