### Part 1: Beginner Exercises

**Strings & Printing**

1. **Hello, World!**: Print the string "Hello, World!".
2. **User Input:** User Input and Greeting: Ask the user for their name and print a personalized greeting.

**Numbers, Operators, & Conditionals**

1. **Simple Calculator:** Create a program that takes two numbers as input and performs addition, subtraction, multiplication, and division.

**Conditionals & Modulo Operator**

1. Even or Odd: Check if a given number is even or odd.

**String Slicing**

1. String Reversal: Write a function that reverses a given string.

**Mathematical Operations & Variables**

1. Area of a Circle: Calculate the area of a circle given its radius.

**Functions & Mathematical Formulas**

1. Temperature Converter: Convert a temperature from Celsius to Fahrenheit and vice versa.

**Conditional Logic**

1. **Largest of Three Numbers:** Find the largest number among three user-provided numbers.

**Strings & Loops**

1. **Vowel Counter:** Count the number of vowels in a given string.

**Loops**

1. Factorial Calculator: Calculate the factorial of a number using a loop.
2. Loops & Sequences
3. 11) Fibonacci Sequence: Generate the first 'n' numbers of the Fibonacci sequence.
4. String Manipulation
5. 12) Palindrome Checker: Check if a string is a palindrome (reads the same forwards and backward).
6. Lists & Loops
7. 13) List Sum: Calculate the sum of all elements in a list.
8. Lists, Loops, & Conditionals
9. 14) Find Maximum in List: Find the largest element in a list without using the built-in max() function.
10. Lists & Counting
11. 15) Count List Elements: Count the occurrences of a specific element in a list.
12. Lists & Sets
13. 16) Unique Elements: Create a new list containing only the unique elements from an original list.
14. Dictionaries
15. 17) Dictionary Basics: Create a dictionary representing a person (name, age, city) and print its values.
16. String Methods
17. 18) Word Count: Count the number of words in a sentence.
18. Mathematical Calculations
19. 19) Simple Interest Calculator: Calculate simple interest given principal, rate, and time.
20. Random Module & While Loops
21. 20) Guess the Number Game: Generate a random number and have the user guess it, providing "higher" or "lower" hints.
22. Loops & String Formatting
23. 21) Multiplication Table: Print the multiplication table for a given number.
24. Functions, Loops, & Logic
25. 22) Check for Prime Number: Write a function to check if a number is prime.
26. String Indexing & Slicing
27. 23) String Slicing: Given a string, print the first 3 characters and the last 3 characters.
28. List Comprehensions
29. 24) List Comprehension: Use list comprehension to create a list of squares of numbers from 1 to 10.
30. Advanced Loops & Conditionals
31. 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

1. File I/O (Reading & Writing)
2. 26) Read and Write to File: Write a program to read content from a text file and write it to a new file.
3. File I/O & String Splitting
4. 27) Count Lines in File: Count the number of lines, words, and characters in a text file.
5. CSV Module
6. 28) CSV File Reader: Read data from a CSV file and print it in a formatted way.
7. JSON Module
8. 29) JSON Data Handling: Read data from a JSON file, parse it, and access specific values.
9. Object-Oriented Programming (Classes)
10. 30) Basic Class: Create a Car class with attributes like make, model, and year, and methods like start\_engine().
11. Object-Oriented Programming (Inheritance)
12. 31) Inheritance: Create an ElectricCar class that inherits from the Car class and adds a battery\_size attribute.
13. Error & Exception Handling
14. 32) Exception Handling: Modify the Simple Calculator (exercise 3) to handle ZeroDivisionError and ValueError.
15. Dictionaries & File I/O
16. 33) Contact Book: Create a simple command-line contact book application using a dictionary to store contacts. Save and load contacts from a file.
17. Lists & File I/O
18. 34) To-Do List Application: A command-line to-do list where users can add, view, and delete tasks. Persist tasks to a file.
19. Algorithms & String Manipulation
20. 35) Caesar Cipher: Implement the Caesar cipher for basic string encryption and decryption.
21. Random Module & String Constants
22. 36) Password Generator: Create a program that generates a random password of a specified length, including a mix of letters, numbers, and symbols.
23. APIs & requests Library
24. 37) Weather API Client: Use the requests library to fetch weather data for a given city from a free weather API.
25. Web Scraping (requests & BeautifulSoup)
26. 38) Web Scraper: Use requests and BeautifulSoup to scrape and print all the headlines from a news website's homepage.
27. Lambda Functions
28. 39) Lambda Functions: Rewrite a simple function (e.g., one that doubles a number) using a lambda function.
29. Functional Programming
30. 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.
31. Recursion
32. 41) Recursive Factorial: Re-implement the factorial calculator (exercise 10) using recursion.
33. Random Module
34. 42) Dice Rolling Simulator: Simulate rolling two dice and display the results.
35. Game Logic & Loops
36. 43) Hangman Game: Create a command-line Hangman game.
37. Game Logic & 2D Lists
38. 44) Tic-Tac-Toe Game: Implement a two-player Tic-Tac-Toe game on the command line.
39. OS Module
40. 45) Directory Tree Lister: Write a script that lists all files and subdirectories within a given directory.
41. requests Library & Binary File I/O
42. 46) Image Downloader: Write a script that downloads an image from a URL and saves it locally.
43. Time Module
44. 47) Simple Stopwatch: Create a command-line stopwatch to measure elapsed time.
45. State Management & Dictionaries
46. 48) Text-Based Adventure Game: Create a simple adventure game where the user makes choices to navigate through a story.
47. APIs & requests Library
48. 49) URL Shortener: Connect to a URL shortener API (like tinyurl) to create a short URL for a given long URL.
49. Regular Expressions (RegEx)
50. 50) Regular Expression Validator: Write a function that uses regex to validate if a string is a valid email address.

### Part 3: Project Ideas

1. Once you're comfortable with the exercises, you can tackle these larger projects to build your portfolio.
2. **AI-Focused Projects**
3. **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.
4. **Simple Chatbot**: Create a rule-based chatbot that can answer predefined questions. For a more advanced version, integrate with a platform like Dialogflow.
5. **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.
6. **Cybersecurity-Focused Projects**
7. **Network Port Scanner**: A tool that scans a target IP address to identify open ports. This is a fundamental tool in network reconnaissance.
8. **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.
9. **Password Strength Checker**: A tool that analyzes a password to determine its strength based on criteria like length, character types, and common patterns.
10. **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.