Day 3: Variables and Data Types
Write a Python program to declare a variable, assign a value to it, and print it.
Create a variable for each data type (int, float, string, boolean) and print their types.
Define a list of 5 elements and access the first and last elements.
Create a tuple with three values and try to modify one of them. What happens?
Create a dictionary with three key-value pairs and print the value of a specific key.
Convert a string containing numbers into an integer using type casting.

Day 4: Lists   Write a program to create a list and add an element using append().
Remove a specific element from a list using remove().
Write a program to demonstrate slicing on a list.

Day 5: Tuples and Sets
Create a tuple and try to change one of its elements. What happens?
Create two sets and perform union and intersection operations.

Day 6: Dictionaries
Create a dictionary and print all the keys, values, and items.
Update a value in a dictionary and print the modified dictionary.

Day 7: String Handling
Write a program to take a user’s name as input and print a greeting message.
Write a program to format a string using f-strings.

Day 8: Operators
Write a program that takes two numbers as input and performs all arithmetic operations on them.
Demonstrate the use of the in operator with a list.

Day 9: Control Flow & Loops
Write a program that checks if a number is positive, negative, or zero.
Write a Python program to find the sum of numbers from 1 to 10 using a for loop.
Write a Python program that prints numbers from 1 to 5 but skips number 3 using continue.

Day 10: Functions
Write a function that takes two numbers and returns their sum.
Write a function that prints "Hello, World!" when called.

Day 11: Modules and Packages
Write a Python program that imports the math module and calculates the square root of a number.
Create a custom module with a function that returns the square of a number.

Day 12: File Handling
Write a program to open a file, write "Hello, Python!" into it, and then read the content.

Day 13: Exception Handling
Write a program that handles division by zero using a try-except block.

Day 14: OOP Basics
Create a class Person with attributes name and age, and create an object of this class.

Day 15: OOP Advanced
Create a class Vehicle with attributes brand and model. Then, create a subclass Car that inherits from Vehicle and adds an attribute fuel_type.Create an object of Car and print all its attributes

Day 16: JSON
Write a Python program to convert a dictionary into a JSON string.

Day 17: Recursion
Write a recursive function to calculate the factorial of a number.

Day 18:
Lambda Functions & List Comprehensions
Write a lambda function to add two numbers.
Use list comprehension to generate a list of even numbers from 1 to 10.

Day 19: Advanced Python
Write a decorator that prints "Function Executed" before calling the actual function.

Day 20: Regular Expressions
Write a Python program that checks if an email is valid using regex.