C++ Course: https://youtu.be/EAR7De6Goz4?si=h2mFhJTBAQcGAN6Y

C Course: https://youtu.be/irqbmMNs2Bo?si=MWYOB_T0scvnJoib

Python Course: https://youtu.be/UrsmFxEIp5k?si=tlOGLpYK2AcFkdHU

Java Course: https://youtube.com/playlist?list=PLsyeobzWxl7pe_IiTfNyr55kwJPWbgxB5&si=afA7Cy5Gobm_P2ge

Day 1

Tasks:

Introduction to programming

Setup development environment

Basic syntax

Printing to console

Practice Problems:

Write a program to print "Hello, World!".

Write a program to print your name.

Write a program to print the current date and time.

Day 2

Tasks:

Variables and Data Types

Practice Problems:

Write a program to declare variables of different types and print their values.

Write a program to swap the values of two variables.

Write a program to find the size of different data types.

Day 3

Tasks:

Operators (Arithmetic, Relational, Logical)

Practice Problems:

Write a program to perform arithmetic operations and print the results.

Write a program to compare two numbers and print the larger one.

Write a program to check if a number is even or odd using logical operators.

Day 4

Tasks:

Control Structures (if, else, switch)

Practice Problems:

Write a program to find the largest of three numbers using if-else.

Write a program to grade students based on their marks using switch-case.

Write a program to check if a number is positive, negative, or zero.

Day 5

Tasks:

Loops (for, while, do-while)

Practice Problems:

Write a program to print the first 10 natural numbers using a loop.

Write a program to calculate the factorial of a number using a loop.

Write a program to print the Fibonacci series up to a specified number.

Day 6

Tasks:

Arrays

Practice Problems:

Write a program to find the sum and average of elements in an array.

Write a program to find the largest element in an array.

Write a program to reverse the elements of an array.

Day 7

Tasks:

Functions

Practice Problems:

Write a program to create a function that calculates the factorial of a number.

Write a program to create a function that checks if a number is prime.

Write a program to create a function that returns the maximum of three numbers.

Day 8

Tasks:

Pointers

Practice Problems:

Write a program to swap two numbers using pointers.

Write a program to find the length of a string using pointers.

Write a program to sort an array using pointer notation.

Day 9

Tasks:

Strings

Practice Problems:

Write a program to reverse a string.

Write a program to check if a string is a palindrome.

Write a program to count the number of vowels and consonants in a string.

Day 10

Tasks:

Structures

Practice Problems:

Write a program to create a structure for a student and print their details.

Write a program to create an array of structures to store information about multiple students.

Write a program to sort an array of structures by a specific field (e.g., student marks).

Day 11

Tasks:

File Handling (reading and writing files)

Practice Problems:

Write a program to read from a file and display its contents.

Write a program to write data to a file.

Write a program to copy the contents of one file to another.

Day 12

Tasks:

Dynamic Memory Allocation (malloc, calloc, free)

Practice Problems:

Write a program to dynamically allocate memory for an array and find its sum.

Write a program to dynamically allocate memory for a matrix and perform matrix addition.

Write a program to release dynamically allocated memory and avoid memory leaks.

Day 13

Tasks:

Preprocessor Directives

Practice Problems:

Write a program using #define to create a constant value.

Write a program using conditional compilation to include/exclude parts of the code.

Write a program to include a custom header file.

Day 14

Tasks:

Recursion

Practice Problems:

Write a program to find the GCD of two numbers using recursion.

Write a program to calculate the factorial of a number using recursion.

Write a program to solve the Tower of Hanoi problem.

Day 15

Tasks:

Practice and Project

Practice Problems:

Write a program to implement a simple calculator using all the concepts learned.

Write a program to manage a simple database (e.g., student records) using file handling and structures.

Write a program to solve a real-world problem (e.g., inventory management) using dynamic memory allocation, structures, and file handling.