Selfcode Academy

www.selfcode.in

Practice Sheet (C and C++ Programming)

This practice sheet contains practical and theoretical questions based on the topics covered in the video lectures. Each lecture includes two practical questions and one theoretical question.
If you ever find that the questions on the practice sheet don't seem directly related to the topic covered in the video lecture or if a question seems out of context, please keep in mind that you will be studying that particular topic in upcoming classes. That's precisely why we have included this practice sheet at the end, so that each question becomes familiar to you and aids in your learning process.

Section 1: Programming with C

Lecture 2: Introduction to C programming

Practical Question: Write a C program to display "Hello, World!" on the console.
Practical Question: Write a C program to take two numbers as input from the user and print their sum.
Theory Question: Explain the basic structure of a C program.

Lecture 3: Variables and Datatypes: (Part 1)

Practical Question: Write a C program to declare and initialize variables of different data types (int, float, char) and display their values.
Practical Question: Write a C program to swap the values of two variables without using a temporary variable.
Theory Question: Explain the concept of variables and data types in C programming.

Lecture 4: Variables and Datatypes: (Part 2)

Practical Question: Write a C program to calculate the area of a rectangle using user-input values for length and width.
Practical Question: Write a C program to convert temperature from Celsius to Fahrenheit.
Theory Question: Differentiate between local variables and global variables in C.

Lecture 5: Type Conversion

Practical Question: Write a C program to take an integer as input and convert it into its equivalent character.
Practical Question: Write a C program to perform arithmetic operations on mixed data types (int and float).
Theory Question: Explain the concept of type conversion in C programming.

Lecture 6: Constants

Practical Question: Write a C program to calculate the area of a circle using a constant value for pi.
Practical Question: Write a C program to define and use symbolic constants for maximum and minimum values.
Theory Question: What is the difference between a constant and a variable in C?

Lecture 7: Operators and Expressions

Practical Question: Write a C program to perform arithmetic operations on two numbers and display the results.
Practical Question: Write a C program to calculate the area of a triangle using the values of base and height entered by the user.
Theory Question: Explain the different types of operators in C programming.

Lecture 8: Control Structures

Practical Question: Write a C program to check whether a number entered by the user is positive, negative, or zero.
Practical Question: Write a C program to determine whether a character entered by the user is a vowel or consonant.
Theory Question: Explain the concept of control structures in C programming.

Lecture 9: Switch Statements

Practical Question: Write a C program to display the name of a day of the week based on the user-inputted number (1-7).
Practical Question: Write a C program to calculate the grade of a student based on their percentage using switch statements.
Theory Question: What is the purpose of using switch statements in C? Provide an example.

Lecture 10: For Loops

Practical Question: Write a C program to print all even numbers between 1 and 50 using a for loop.
Practical Question: Write a C program to calculate the factorial of a number using a for loop.
Theory Question: Explain the structure and working of a for loop in C programming.

Lecture 11: Nested for loop, While loop, and Do-while loop

Practical Question: Write a C program to display a pyramid pattern using nested for loops.
Practical Question: Write a C program to find the sum of digits of a number using a while loop.
Theory Question: Compare and contrast the while loop and do-while loop in C.

Lecture 12: Arrays

Practical Question: Write a C program to find the largest element in an array.
Practical Question: Write a C program to calculate the average of elements in an array.
Theory Question: Explain the concept of arrays and their use in C programming.

Lecture 13: Strings

Practical Question: Write a C program to concatenate two strings entered by the user.
Practical Question: Write a C program to reverse a string using a loop.
Theory Question: What is a string in C? Explain the functions used to manipulate strings.

Lecture 14: Functions: Part 1

Practical Question: Write a C program to find the factorial of a number using a user-defined function.
Practical Question: Write a C program to check whether a number is prime or not using a user-defined function.
Theory Question: Explain the concept of functions in C programming and their advantages.

Lecture 15: Functions: Part 2

Practical Question: Write a C program to calculate the power of a number using a recursive function.
Practical Question: Write a C program to find the maximum of three numbers using a user-defined function.
Theory Question: Discuss the difference between a function declaration and a function definition in C.

Lecture 16: Pointers: Part 1

Practical Question: Write a C program to swap the values of two variables using pointers.
Practical Question: Write a C program to find the length of a string using a pointer.
Theory Question: Explain the concept of pointers in C and their importance in programming.

Lecture 17: Pointers: Part 2

Practical Question: Write a C program to pass an array to a function and display its elements using pointers.
Practical Question: Write a C program to allocate memory dynamically for an integer and store its value using a pointer.
Theory Question: What is the difference between pass-by-value and pass-by-reference in C? Explain with examples.

Lecture 18: Files

Practical Question: Write a C program to read data from a text file and display it on the console.
Practical Question: Write a C program to write data to a binary file.
Theory Question: How are files handled in C? Explain the different file modes used in file handling.

Lecture 19: Hands-on Practice: Questions: Part 1

Practical Question: Write a C program to find the sum of all even numbers between 1 and 100 using a loop.
Practical Question: Write a C program to reverse the digits of a number using a while loop.
Theory Question: Explain the concept of recursion and its advantages in programming.

Lecture 20: Hands-on Practice: Questions: Part 2

Practical Question: Write a C program to check whether a string is a palindrome or not.
Practical Question: Write a C program to sort an array of integers in ascending order using a user-defined function.
Theory Question: What is the purpose of header files in C programming? Provide examples of commonly used header files.

Section 2 : Programming with C++

Lecture 22: Introduction to C++

Practical Question: Write a C++ program to display "Hello, World!" on the console.
Practical Question: Write a C++ program to take two numbers as input from the user and print their sum.
Theory Question: Explain the key features and advantages of C++ programming.

Lecture 23: Variables

Practical Question: Write a C++ program to declare and initialize variables of different data types (int, float, char) and display their values.
Practical Question: Write a C++ program to swap the values of two variables without using a temporary variable.
Theory Question: Discuss the scope and lifetime of variables in C++.

Lecture 24: Data Types

Practical Question: Write a C++ program to calculate the area of a rectangle using user-input values for length and width.
Practical Question: Write a C++ program to convert temperature from Celsius to Fahrenheit.
Theory Question: Explain the different data types available in C++.

Lecture 25: User-Input

Practical Question: Write a C++ program to take user-input for a student's name, roll number, and marks in three subjects, and display the details.
Practical Question: Write a C++ program to take user-input for two numbers and perform arithmetic operations on them.
Theory Question: How can user input be obtained in C++? Explain with examples.

Lecture 26: Operators

Practical Question: Write a C++ program to perform arithmetic operations on two numbers and display the results.
Practical Question: Write a C++ program to calculate the area of a triangle using the values of base and height entered by the user.
Theory Question: Discuss the different types of operators in C++.

Lecture 27: Control Structure

Practical Question: Write a C++ program to check whether a number entered by the user is positive, negative, or zero.
Practical Question: Write a C++ program to determine whether a character entered by the user is a vowel or consonant.
Theory Question: Explain the concept of control structures in C++.

Lecture 28: Switch Statements

Practical Question: Write a C++ program to display the name of a day of the week based on the user-inputted number (1-7).
Practical Question: Write a C++ program to calculate the grade of a student based on their percentage using switch statements.
Theory Question: What is the purpose of using switch statements in C++? Provide an example.

Lecture 29: Loops

Practical Question: Write a C++ program to print all even numbers between 1 and 50 using a loop.
Practical Question: Write a C++ program to calculate the factorial of a number using a loop.
Theory Question: Explain the structure and working of loops in C++.

Lecture 30: Questions on Loops

Practical Question: Write a C++ program to display a pyramid pattern using nested loops.
Practical Question: Write a C++ program to find the sum of digits of a number using a loop.
Theory Question: Compare and contrast the different types of loops in C++.

Lecture 31: Break and Continue

Practical Question: Write a C++ program to print the numbers from 1 to 10, skipping the number 7 using the continue statement.
Practical Question: Write a C++ program to find the first prime number between 1 and 100 using the break statement.
Theory Question: Explain the use of the break and continue statements in C++.

Lecture 32: Array

Practical Question: Write a C++ program to find the largest element in an array.
Practical Question: Write a C++ program to calculate the sum of elements in an array.
Theory Question: Discuss the concept of arrays in C++ and their importance in programming.

Lecture 33: String

Practical Question: Write a C++ program to concatenate two strings entered by the user.
Practical Question: Write a C++ program to reverse a string using a loop.
Theory Question: What is a string in C++? Explain the functions used to manipulate strings.

Lecture 34: Functions

Practical Question: Write a C++ program to find the factorial of a number using a user-defined function.
Practical Question: Write a C++ program to check whether a number is prime or not using a user-defined function.
Theory Question: Explain the concept of functions in C++ programming and their advantages.

Lecture 35: Pointers

Practical Question: Write a C++ program to swap the values of two variables using pointers.
Practical Question: Write a C++ program to find the length of a string using a pointer.
Theory Question: Explain the concept of pointers in C++ and their importance in programming.

Lecture 36: Structure & Unions

Practical Question: Write a C++ program to define a structure representing a student and display its details.
Practical Question: Write a C++ program to define a union representing a person's name and display the name in different formats.
Theory Question: Discuss the use of structures and unions in C++.

Lecture 37: Files

Practical Question: Write a C++ program to read data from a text file and display it on the console.
Practical Question: Write a C++ program to write data to a binary file.
Theory Question: How are files handled in C++? Explain the different file modes used in file handling.

Lecture 38: Classes & Objects

Practical Question: Write a C++ program to create a class representing a rectangle and calculate its area.
Practical Question: Write a C++ program to create a class representing a bank account and perform deposit and withdrawal operations.
Theory Question: Explain the concepts of classes and objects in C++.

Lecture 39: Class Methods

Practical Question: Write a C++ program to create a class representing a calculator and perform arithmetic operations using class methods.
Practical Question: Write a C++ program to create a class representing a book and display its details using class methods.
Theory Question: Discuss the use of class methods in C++.

Lecture 40: Object-oriented Programming

Practical Question: Write a C++ program to create a class representing a car and simulate its movement using object-oriented programming concepts.
Practical Question: Write a C++ program to create a class representing a bank and perform account-related operations using object-oriented programming.
Theory Question: Explain the concept of object-oriented programming (OOP) and its advantages over procedural programming.