

# C++ Programming Practice Questions

## Data Types

Write a program to display the sizes of all fundamental data types in C++.

Declare variables of all basic data types and initialize them with values.

Create a program that takes input from the user for each data type and displays the entered values.

Write a program to calculate the area of a circle using float and double data types.

Use char data type to print ASCII values of characters entered by the user.

Create a program to demonstrate the use of bool data type.

## Operators

Write a program to demonstrate arithmetic operators.

Implement a calculator using basic operators (+, -, \*, /, %).

Demonstrate the use of increment and decrement operators.

Create a program to show the difference between post-increment and pre-increment.

Write a program to swap two numbers.

Write a program to swap two numbers using arithmetic operator.

Use relational operators to compare two user-input numbers.

Implement a program using logical operators to validate user credentials.

Demonstrate the use of the conditional (ternary) operator.

Use assignment operators (+=, -=, \*=, etc.) in a program.

Write a program to find the maximum of two numbers using the conditional operator.

## If-Else Statements

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

Implement a grading system based on user input marks 80-90(A), 60-80(B), 40-60(C), 0-40(D).

Check if a number is even or odd using an if-else statement.

Write a program to find the largest of three numbers.

Implement a program to check if a year is a leap year.

Write a program to determine the eligibility to vote based on age.

Create a program to check if a character is a vowel or consonant.

Implement a simple number guessing game with if-else statements.

Write a program to validate user input based on conditions Ex ATM.

Create a program to calculate the discount based on purchase amount using conditions 0-1000(5%), 1000-5000(10%), 5000-10000(15%), 10000-30000(20%) and more then 50000(30%).

## Switch Case

Write a program to display the day of the week based on user input.

Implement a basic calculator using a switch case.

Implement a menu-driven program for basic string operations.

Create a program to display the name of a country based on the ISO country code **Ex: India(IND), Australia (AUS), Colombia(COL), Germany(DEU), Mexico(MEX).**

Write a program to classify animals based on user input using a switch **Ex: DOG(wow, wow), CAT(meow,meow).**

## Loops

Print the first 10 natural numbers using a loop.

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

Implement a program to find the sum of the first N natural numbers using a while loop.

Implement a program to find all numbers divisible by 3 and 5 within a range.

Write a program to calculate the sum of squares of the first N natural numbers.

Print the multiplication table of a number.

Write a program to check if a number is a prime number.

Implement a program to reverse a number.

Generate Fibonacci series up to a given term using loops.

Implement a program to count the number of digits in a number.

Write a program to calculate the sum of digits of a number.

Check if a number is a palindrome using loops.

Print all prime numbers within a given range.

Write a program to find the LCM of two numbers.

Write a program to find the HCF of two numbers.

Create a program to calculate the power of a number using loops.

Print all Armstrong numbers within a range.

Write a program to find the sum of even and odd numbers separately in a range.

Half Pyramid of \*

```
*  
* *  
* * *  
* * * *  
* * * * *
```

Inverted half pyramid of \*

```
* * * * *  
* * * *  
* * *  
* *  
*
```

Half Pyramid of Numbers

```
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5
```

Half Pyramid of Alphabets

```
A
```

B B

C C C

D D D D

E E E E E

Full Pyramid of \*

```
      *
    * * *
  * * * * *
* * * * * *
* * * * * * *
* * * * * * * *
```

## Arrays

1. Write a program to take input and display elements of an array.
2. Find the largest elements in an array.
3. Find the smallest elements in an array.
4. Implement a program to calculate the sum of array elements.
5. Implement a program to sort an array in ascending order.
6. Implement a program to sort an array in decending order.
7. Find the second largest element in an array.
8. Find the Kth largest element in an array.
9. Remove duplicates from an array.
10. Write a program to reverse an array.
11. Implement a program to find the frequency of each element in an array.
12. Check if an array is sorted in ascending order.
13. Check if an array is sorted in descending order.
14. Write a program to find the intersection of two arrays.
15. Implement a program to shift array elements to the left or right.

16. Create a program to rotate an array.
17. Write a program to find the missing number in a series.
18. Implement a program to count positive and negative numbers in an array.
19. Write a program to check if two arrays are equal.
20. Find the maximum product of two elements in an array.
21. Write a program to count the number of zeros in an array.
22. Write a program to shift zeros in the last of the array.

## Functions

1. Write a program to demonstrate a simple function that prints "Hello, World!".
2. Create a function to add two numbers and return the result.
3. Implement a program with a function to find the maximum of three numbers.
4. Write a function to calculate the factorial of a number.
5. Create a function to check whether a number is prime or not.
6. Write a program to demonstrate function overloading with different types of parameters.
7. Implement a program to calculate the area of a rectangle, circle, and triangle using separate functions.
8. Create a recursive function to calculate the nth Fibonacci number.
9. Write a function to reverse a number.
10. Create a function to swap two numbers using call by value and call by reference.

## Strings

1. Write a program to take input and display a string.
2. Check if a string is a palindrome.
3. Count the number of vowels and consonants in a string.
4. Reverse a string without using inbuilt functions.
5. Write a program to concatenate two strings.

6. Implement a program to find the length of a string.
7. Count the frequency of each character in a string.
8. Write a program to compare two strings.
9. Implement a program to reverse the order of words in a string.
10. Check if two strings are anagrams.

## **OOP Concepts**

1. Write a program to create a class and call its functions.
2. Implement a program to demonstrate constructors and destructors.
3. Write a program to demonstrate inheritance.
4. Implement a program to demonstrate method overloading.
5. Write a program to demonstrate method overriding.
6. Create a program to implement a basic calculator using classes.
7. Implement a program to demonstrate encapsulation.
8. Demonstrate the use of static members in a class.
9. Implement a program to demonstrate operator overloading.
10. Write a program to demonstrate friend functions.
11. Implement a program to demonstrate abstract classes.
12. Write a program to demonstrate polymorphism.
13. Write a program to demonstrate multiple inheritance.
14. Implement a program to demonstrate virtual functions.
15. Write a program to demonstrate runtime polymorphism.

