c++

1. Variables:
Create a program that calculates the area of a rectangle using variables for length and width.

2. Strings - Working with Text:
Write a program that takes a user's name as input and outputs a personalised greeting.

3. User Input:
Develop a program that takes two numbers from the user, adds them, and displays the result.

4. Binary Numbers and Computer Memory:
Create a program that converts a decimal number to its binary representation.

5. Integer Types:
Write a program that checks if a given integer is even or odd.

6. Floating Point Types:
Develop a program that calculates the square root of a given positive number.

7. Other Types: Char and Bool:
Create a program that checks if a given character is a vowel or consonant.

8. If:
Write a program that checks if a given number is positive.

9. If-Else:
Develop a program that determines whether a given number is positive or negative.

10. If-Else If-Else:
Create a program that assigns grades (A, B, C, etc.) based on a given numerical score.

11. Comparing Floats:
Write a program that compares two floating-point numbers and checks if they are equal within a certain tolerance.

12. C++ Conditions:
Develop a program that takes a number from the user and displays whether it's positive, negative, or zero.

13. While Loops:
Create a program that prints the multiplication table of a given number using a while loop.

14. Do-While Loops:
Write a program that takes numbers from the user until they enter zero, and then displays the sum of the entered numbers.

15. "For" Loops:
Develop a program that calculates the factorial of a given positive integer using a for loop.

16. Break and Continue:
Create a program that takes numbers from the user until they enter a negative number. Display the sum of the entered positive numbers.

17. Arrays - Lists of Data:
Write a program that finds the maximum and minimum values in an array of integers.

18. Multidimensional Arrays:
Develop a program that calculates the sum of elements in a 2D array.

19. Sizeof and Arrays:
Create a program that calculates and displays the size of an integer array.

20. Size of Multidimensional Arrays:
Write a program that calculates and displays the size of a 2D array.

21. Switch:
Develop a program that takes a number from the user and displays the corresponding day of the week using a switch statement.

22. Functions:
Create a simple program that defines a function to display a welcome message.

23. Return Values:
Write a program that defines a function to calculate the square of a number and returns the result.

24. Function Parameters:
Develop a program that defines a function to calculate the sum of two numbers and takes the numbers as function parameters.

25. Headers and Prototypes:
Create a program with a separate header file and source file. Define a function in the header file, and then implement and use it in the source file.

26. Classes and Data Members:
Create a class Person with data members for name and age. Instantiate an object of the class and display its information.

27. Constructors and Destructors:
Define a class Book with a constructor and a destructor. Instantiate an object of the class to observe constructor and destructor calls.

28. Getters and Setters:

Enhance the Person class with getter and setter methods for name and age. Use these methods to access and modify the object's data members.

29. String Streams:
Develop a program that uses string streams to concatenate a name and age into a single sentence.

30. Overloading Constructors:
Extend the Person class with multiple constructors that allow initializing objects with different sets of information.

31. The "this" Keyword:
Create a class Rectangle with data members for width and height. Implement a method that uses the this keyword to calculate the area.

32. Constructor Initialization Lists:
Modify the Book class to use constructor initialization lists to set initial values for data members.

33. Pointers:
Write a program that declares a pointer variable and assigns it the address of an integer variable. Print both the value and the address of the integer variable using the pointer.

34. Arithmetic:
Develop a program that demonstrates pointer arithmetic by incrementing a pointer to an integer and printing its value.

35. Pointers and Arrays:
Create a program that uses a pointer to iterate through an array of integers and display their values.

36. Pointer Arithmetic:
Enhance the previous program by performing pointer arithmetic to calculate the sum of elements in an array.

37. Char Arrays:
Write a program that uses a char array to store a string and then displays the string.

38. Reversing a String:
Develop a program that takes a string as input and outputs its reverse using pointer arithmetic.

39. References:
Create a program that demonstrates the use of references to modify the value of a variable.