Question bank for c++

Introduction to OOPs

1. Define Object-Oriented Programming. How is it different from procedural programming?
2. Explain the four pillars of OOP with suitable examples.
3. What are the advantages of OOP? Illustrate with examples.
4. Differentiate between a class and an object with examples.

OOPs Concepts

5. Explain the concept of encapsulation and its importance in programming.
6. What is polymorphism? Discuss its types with examples.
7. Describe abstraction with a real-life example. How is it implemented in C++?
8. Define inheritance and explain how it helps in code reusability.

Introduction to C++

9. Discuss the key features of C++ that make it an OOP language.
10. Compare C++ with C. What makes C++ more powerful than C?

Declaration, Initialization, and Operators

11. Explain the rules for variable declaration and initialization in C++ with examples.
12. Discuss the use of arithmetic and relational operators in C++ with examples.
13. Write a short note on the scope and lifetime of variables in C++.
14. What is the difference between pre-increment and post-increment operators in C++?

Decision Control: If-Else Statements

15. Explain the use of nested if-else statements with an example.
16. How to find the largest of three numbers using if-else statements. Demonstrate with the help of code.

Iteration Control: Loops

17. Discuss the differences between for, while, and do-while loops with examples.
18. Write a program to calculate the sum of the first N natural numbers using a loop.

Array, Types of Arrays, Operations

19. Explain the concept of arrays and their significance in programming.
20. How to sort an array of integers.
21. Differentiate between single-dimensional and multi-dimensional arrays with examples.
22. Discuss the advantages and disadvantages of using arrays.

String Declaration and Manipulation

23. How to reverse a string using a function.
24. Explain the different methods of string initialization in C++.
25. Explain string functions with examples?

Function Declaration, Definition, and Parameters

26. Differentiate between call-by-value and call-by-reference.
27. Explain the difference between function declaration and function definition in C++.
28. Discuss the use of default arguments in functions with examples.

Pointers

29. Explain the use of pointers to access array elements.
30. What do you mean by pointers? Explain with the help of an example?

Reference Variables as Function Parameters

31. Explain the concept of reference variables with an example.
32. How do reference variables differ from pointers in C++?

New and Delete Operators

33. Explain dynamic memory allocation using new and delete. Explain with the help of an example
34. How to create an array using the new and delete operator?

Class Definition and Object Creation

35. Explain the constructor in detail. Create a class representing a student and initialize it with a constructor.
36. How do objects interact with member functions in a class? Explain with an example.

Constructors, Destructors, and Scope Resolution Operator

37. Explain parameterized constructors with the help of an example
38. What is the scope resolution operator? Explain its usage in accessing global variables.
39. What is the role of a destructor in releasing resources?

Access Specifiers and Getter/Setter Methods

40. Explain the purpose of access specifiers in C++ with examples.
41. What is the use of getter and setter methods in C++.

Inheritance and Access Specifiers in Inheritance

42. Explain single inheritance with proper access specifiers.
43. Discuss the difference between public, private, and protected inheritance.

Types of Inheritance

44. Explain multiple inheritance with an example program.
45. Explain the concept of hierarchical inheritance with the help of an example.

Function Overloading and Virtual Functions

46. Explain function overloading in C++ with the help of an example.
47. Explain the role of virtual functions in achieving runtime polymorphism with an example.

File I/O Operations

48. How to read and write data to a file in C++ using the File Handling concept.
49. Discuss the difference between ifstream, ofstream, and fstream in file handling.

Try-Catch Blocks and Exceptions

50. Explain try-catch blocks for exception handling with the help of an example?

Function Templates and Class Templates

51. What is the use of a function template?
52. Explain the role of class templates in generic programming with an example.