C++ Interview Questions

1. Basic C++ Concepts

1. **What is C++ and how does it differ from C?**
2. **What are the main features of C++?**
3. **What are the basic data types in C++?**
4. **What is the difference between a variable and a constant?**
5. **What is the significance of the main() function?**
6. **How do you compile and run a C++ program?**
7. **What are header files and why are they used?**
8. **What is the role of the preprocessor?**
9. **What is a namespace and why is it important?**
10. **What are the basic input and output operations in C++?**

---

2. Control Structures and Functions

11. **How do loops (for, while, do-while) work in C++?**
12. **What is the difference between if-else and switch statements?**
13. **How do you declare and define a function in C++?**
14. **What is function overloading?**
15. **What is function overriding and how does it work?**
16. **What are default arguments in functions?**
17. **How does recursion work in C++?**
18. **What is the inline function and when should it be used?**
19. **What is a function pointer?**
20. **What is the use of the ternary operator?**

---

3. Object-Oriented Programming (OOP) Concepts

21. **What are classes and objects in C++?**
22. **What is encapsulation and how is it achieved?**
23. **What is inheritance? List the different types of inheritance.**
24. **What is polymorphism and how is it implemented in C++?**
25. **What are virtual functions and why are they used?**

26. What is an abstract class and how do you declare one?
27. What is an interface in C++ (using pure virtual functions)?
28. What is the role of the this pointer?
29. What is the purpose of the super keyword (or its equivalent in C++)?
30. What is multiple inheritance and what potential issues can arise from it?

---

4. Constructors, Destructors, and Copy Control

31. What is a constructor and how does it differ from a regular method?
32. What is a destructor and when is it called?
33. Explain the concept of copy constructor.
34. What is the assignment operator, and how does it differ from the copy constructor?
35. What is the Rule of Three?
36. What is the Rule of Five in C++11?
37. What is a move constructor and when would you use it?
38. What is a move assignment operator?
39. What is resource management and how does RAII (Resource Acquisition Is Initialization) help?
40. What is a delegating constructor?

---

5. Memory Management and Pointers

41. What is dynamic memory allocation in C++?
42. What is the difference between malloc/free and new/delete?
43. How do pointers work in C++?
44. What is a dangling pointer and how can it be avoided?
45. What is pointer arithmetic?
46. What are references, and how are they different from pointers?
47. What is a null pointer?
48. What is a smart pointer and what types does C++11 provide (e.g., std::unique_ptr, std::shared_ptr)?
49. How do you prevent memory leaks in C++?

50. What is RAII and why is it important for resource management?

---

6. Templates and Generic Programming

51. What are templates in C++?

52. What is the difference between function templates and class templates?

53. How do template specializations work?

54. What are variadic templates?

55. What is the purpose of the typename keyword in templates?

56. How do you implement a generic container class using templates?

57. What are some common pitfalls when using templates?

58. How do compile-time errors in templates differ from runtime errors?

59. What is SFINAE (Substitution Failure Is Not An Error)?

60. How do you constrain templates in C++20 with concepts?

---

7. Standard Template Library (STL)

61. What is the STL and what are its components?

62. What are containers in STL? Give examples (e.g., vector, list, deque).

63. What is the difference between sequence containers and associative containers?

64. What are iterators and how do they work?

65. What is the purpose of algorithms in STL?

66. What is a functor in C++?

67. How do you use std::sort and what are its requirements?

68. What is the difference between std::map and std::unordered_map?

69. What is a lambda expression in C++?

70. How can you perform custom sorting on STL containers?

---

8. Exception Handling and Advanced Topics

71. How does exception handling work in C++?

72. What is the try-catch block and how do you use it?

73. What is the purpose of the throw keyword?

74. What is a custom exception, and how do you create one?
75. What is the function of the noexcept specifier?
76. What is stack unwinding?
77. What are the potential issues with exception safety in C++?
78. How do you handle exceptions in constructors and destructors?
79. What is the difference between standard exceptions and user-defined exceptions?
80. What are some best practices for error handling in C++?

---

9. C++11 and Later Features

81. What are some of the new features introduced in C++11?
82. What is auto type deduction and how does it work?
83. How do lambda expressions improve C++ code?
84. What are range-based for loops?
85. What is the difference between std::move and std::forward?
86. What is a nullptr and why is it used?
87. What are rvalue references and how do they work?
88. What are the benefits of using std::thread for multithreading?
89. What is the purpose of the constexpr specifier?
90. What are user-defined literals?

---

10. Miscellaneous and Best Practices

91. What is name mangling in C++?
92. What is the significance of the virtual table (vtable) and virtual pointer (vptr)?
93. How does multiple inheritance affect memory layout?
94. What is the diamond problem in C++ and how is it solved?
95. What is the difference between shallow copy and deep copy?
96. What are the common pitfalls in C++ programming?
97. How do you implement operator overloading and what are some best practices?
98. What is the significance of the copy-and-swap idiom?
99. What are some techniques to write thread-safe code in C++?

C++ Interview Questions

100. How do you optimize C++ code for performance?