1. What is difference between in C and C++?

|  |  |
| --- | --- |
| C supports procedural programming paradigm for code development. | C++ supports both procedural and object oriented programming paradigms; therefore C++ is also called a hybrid language. |
| C does not support object oriented programming; therefore it has no support for polymorphism, encapsulation, and inheritance. | Being an object oriented programming language C++ supports polymorphism, encapsulation, and inheritance. |
| In C (because it is a procedural programming language), data and functions are separate and free entities. | In C++ (when it is used as object oriented programming language), data and functions are encapsulated together in form of an object. For creating objects class provides a blueprint of structure of the object. |
| In C, data are free entities and can be manipulated by outside code. This is because C does not support information hiding. | In C++, Encapsulation hides the data to ensure that data structures and operators are used as intended. |
| C, being a procedural programming, it is a function driven language. | While, C++, being an object oriented programming, it is an object driven language. |
| C does not support function and operator overloading. | C++ supports both function and operator overloading. |
| C does not allow functions to be defined inside structures. | In C++, functions can be used inside a structure. |
| C does not have namespace feature. | C++ uses NAMESPACE which avoid name collisions.  A namespace is a declarative region that provides a scope to the identifiers (the names of types, functions, variables, etc) inside it. Namespaces are used to organize code into logical groups and to prevent name collisions that can occur especially when your code base includes multiple libraries. All identifiers at namespace scope are visible to one another without qualification. Identifiers outside the namespace can access the members by using the fully qualified name for each identifier. |
| C uses functions for input/output. For example scanf and printf. | C++ uses objects for input output. For example cin and cout. |
| C does not support reference variables. | C++ supports reference variables. |
| C has no support for virtual and friend functions. | C++ supports virtual and friend functions. |
| C provides malloc() and calloc() functions for dynamic memory allocation, and free() for memory de-allocation. | C++ provides new operator for memory allocation and delete operator for memory de-allocation. |
| C does not provide direct support for error handling (also called exception handling) | C++ provides support for exception handling. Exceptions are used for "hard" errors that make the code incorrect. |

1. Why is the size of an empty class not zero in C++?

To give a identification on memory space for the instance created for the class a minimum memory is allocated to the instance.

1. What are C++ access, specifiers ?

<http://www.trytoprogram.com/cplusplus-programming/access-specifiers/#:~:targetText=C%2B%2B%20Access%20Specifiers%20%E2%80%93%20Private%2C%20Public%20and%20Protected&targetText=C%2B%2B%20access%20specifiers%20are%20used,sections%2C%20private%20protected%20and%20public%20.>

1. What are the various OOPs concepts in C++?
2. What are the differences between a C++ struct and C++ class?
3. What is difference between new and malloc?
4. What is a namespace?
5. What is Overriding?
6. How to create .dll in C++ and how to link .dll in your code?
7. What is overloading?
8. How to create and use a reference variable in c++?
9. Can you overload a function based only on whether a parameter is a value or a reference?
10. What is difference between function overloading and Operator Overloading?
11. Can we access private data members of a class without using a member or a friend function?
12. What is the use of inline function?
13. how to access derived class function from base class object without using virtual function?
14. Distinguish between shallow copy and deep copy.
15. Friend class and function in C++
16. what is difference between constructor and destructor?
17. How are .h files loaded and linked with their .c files?
18. What Is Inheritance?
19. What is “this” pointer?
20. What is difference between delete and free?
21. What is encapsulation?
22. What is static Member in c++?
23. What is a pure virtual function in c++?
24. Count the number of words, characters, and lines in a file?
25. Any fundamental difference between source and header files in C?
26. Can a C++ class member function template be virtual?
27. What is the Diamond problem? How can we get around it?
28. What are the advantages of inheritance?
29. why virtual functions cannot be static in c++?
30. How does the compilation/linking process work?
31. Can you explain the order of execution in constructor initialization list?
32. How to make a C++ class whose objects can only be dynamically allocated?
33. Is it possible to have Virtual Constructor? If yes, how? If not, Why not possible?
34. What is a constructor? Is default constructor exist in C++?
35. Can a constructor throw an exception? How to handle the error when the constructor fails?
36. What is the difference between a copy constructor and an overloaded assignment operator?
37. When are copy constructors called in C++?
38. Why copy constructor takes the parameter as a reference in C++?
39. What do you mean by inline function?
40. What is a template function?
41. Can we combine c and C++ code?
42. What is shared pointer in c++?
43. What is typecasting?
44. Why is a pure virtual function initialized by 0?
45. How to create multidimensional arrays using new?
46. When should static\_cast, dynamic\_cast, const\_cast and reinterpret\_cast be used?
47. Can I free() pointers allocated with new?
48. How do I convert an integer to a string in C++?
49. What is the difference between a pointer and a reference?
50. When should I use references, and when should I use pointers?
51. What is the assignment operator in c++?
52. What does the explicit keyword mean?
53. When do we need to create virtual destructor in our class?
54. What is Polymorphism?
55. What is the effect of extern “C” in C++?
56. What are the different types of polymorphism in C++?
57. Can a copy constructor accept an object of the same class as a parameter, in place of reference  
    of the object? If No, why not possible?
58. What about Virtual Destructor?
59. What is conversion constructor?
60. How do you access the static member of a class?
61. When do we use the Initializer List in C++?
62. Is it possible to overload the destructor the class?
63. Which is faster: Stack allocation or Heap allocation
64. What is auto pointer in C++?
65. What is the smart pointer in C++?
66. Can I delete pointers allocated with malloc()?
67. Can I use realloc() on pointers allocated via new?
68. Can I call the destructor explicitly?
69. Can one constructor of a class call another constructor of the same class to initialize the this  
    object?
70. What is the difference between an array and a list?
71. What is a “pure virtual” member function and when it is used?
72. What are VTABLE and VPTR?
73. How to handle the exception in C++?
74. What is a Memory Leak?
75. why static functions cannot access non-static variables?
76. What is a dangling pointer?
77. What is “mutable” keyword in C++?
78. What are the debugging methods you use when came across a problem?
79. STL Containers – What are the types of STL containers?