|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **1. Name the operators that cannot be overloaded.? sizeof, ., .\*, .->, ::, ?  2. What is a dangling pointer? A dangling pointer arises when you use the address of an object after its lifetime is over. This may occur in situations like returning addresses of the automatic variables from a function or using the address of the memory block after it is freed.   3. How do I call a C function from C++? Just declare the C function ``extern "C"'' (in your C++ code) and call it (from your C or C++ code)   4. Is C a subset of C++? In the strict mathematical sense, C isn't a subset of C++. There are programs that are valid C but not valid C++ and even a few ways of writing code that has a different meaning in C and C++. However, C++ supports every programming technique supported by C. Every C program can be written in essentially the same way in C++ with the same run-time and space efficiency. It is not uncommon to be able to convert tens of thousands of lines of ANSI C to C-style C++ in a few hours. Thus, C++ is as much a superset of ANSI C   5. What is inheritance? Inheritance is property such that a parent (or super) class passes the characteristics of itself to children (or sub) classes that are derived from it. The sub-class has the option of modifying these characteristics in order to make a different but fundamentally related class from the super-class.   6. What are instance variables? These represent an object's private memory. They are defined in an object's class.   7. What is a super-class? Given a class, a super-class is the basis of the class under consideration. The given class is defined as a subset (in some respects) of the super-class. Objects of the given class potentially posses all the characteristics belonging to objects of the super-class.   8. What is an object in C++? An object is a package that contains related data and instructions. The data relates to what the object represents, while the instructions define how this object relates to other objects and itself.   9. What is polymorphism? Polymorphism refers to the ability of an object to respond in a logically identical fashion to messages of the same protocol, containing differing types of objects. Consider 1 + 5 and 1 + 5.1. In the former, the message "+ 5" is sent to an object of class integer (1). In the later, the message "+ 5.1" is sent to the same integer object. The form of the message (its protocol) is identical in both cases. What differs is the type of object on the right-hand side of these messages. The former is an integer object (5) while the later is a floating point object (5.1). The receiver (1) appears (to other objects) to respond in the same way to both messages. Internally, however, it knows that it must treat the two types of objects differently in order to obtain the same overall response.   10. What is operator overloading? It is the process of, and ability to redefine the way an object responds to a C++ operator symbol. This would be done in the object's class definition.   11. Define Copy Constructor. A copy constructor is invoked when you initialize a new object of a class using an existing object. This will happen when: • You pass a copy of an object as argument to a function (i.e. when passing by value).  • When you return an object from a function  • Or initialize an object during declaration using another object.  If we don’t specify a copy constructor, the compiler already has a default copy constructor.   12. What is the difference between overriding and overloading?  An overrided method redefines an existing method in a superclass to specify a new behavior. Overloading means that you have multiple methods with the same name but with a different parameter list.** | | |  | | --- | | **13. What is inline function? When you call the function, execution of the program jumps to those instructions, and when the function returns, execution jumps back to the next statement in the calling function. Performance degradation occurs in jumping into and out of functions. When a function is small, the program can be speeded-up using inline functions. The use of inline functions avoids have the program jump into and out of the same function over and over again.If a function is declared with the keyword 'inline', the compiler does not create a real function; it copies the code from the inline function directly into the calling function. No jump is made.   14. List down the advantages of class templates • One C++ Class Template can handle different types of parameters.  • Compiler generates classes for only the used types. If the template is instantiated for int type, compiler generates only an int version for the c++ template class.  • Templates reduce the effort on coding for different data types to a single set of code.  • Testing and debugging efforts are reduced.  15. Write a short notes on recursion with example. Recursion is defined as a function calling itself. It is in some ways similar to a loop because it repeats the same code, but it requires passing in the looping variable and being more careful. Many programming languages allow it because it can simplify some tasks, and it is often more elegant than a loop. void recurse() { recurse(); } void main() { recurse(); }   16. Is it possible to accept command line argument in C++? Yes, In C++ it is possible to accept command line arguments. To do so, you must first understand the full definition of int main(). It actually accepts two arguments, one is number of command line arguments, the other is a listing of the command line arguments. Ex. int main( int argc, char \*argv[]) 17. Explain typecasting. Typecasting is making a variable of one type, such as an int, act like another type, a char, for one single operation. To typecast something, simply put the type of variable you want the actual variable to act as inside parentheses in front of the actual variable. (char)a will make 'a' function as a char.   18. What’s the difference between public, private and protected? • A member (either data member or member function) declared in a private section of a class can only be accessed by member functions and friends of that class  • A member (either data member or member function) declared in a protected section of a class can only be accessed by member functions and friends of that class, and by member functions and friends of derived classes  • A member (either data member or member function) declared in a public section of a class can be accessed by anyone   19. What are some advantages/ disadvantages of using friend functions? Member functions and friend functions are equally privileged. The major difference is that a friend function is called like f(x), while a member function is called like x.f(). Thus the ability to choose between member functions (x.f()) and friend functions (f(x)) allows a designer to select the syntax that is deemed most readable, which lowers maintenance costs. The major disadvantage of friend functions is that they require an extra line of code when you want dynamic binding.   20. What’s the difference between the keywords struct and class in c++? The members and base classes of a struct are public by default, while in class, they default to private.** | |