1. What is inheritance in C++?
   1. A way to hide data
   2. A way to implement polymorphism
   3. **A way to create new classes from existing classes**
   4. A way to handle exceptions
2. Which keyword is used to derive a class from a base class in C++?
   1. base
   2. super
   3. this
   4. **class**
3. Which access specifier is used for private inheritance in C++?
   1. **private**
   2. protected
   3. public
   4. friend
4. In C++, when deriving a class from a base class, which access specifier grants the most visibility to the members of the base class in the derived class?
   1. private
   2. protected
   3. **public**
   4. friend
5. What is the default access specifier while inheriting a class in C++?
   1. **private**
   2. protected
   3. public
   4. friend
6. Which type of inheritance involves deriving a class from multiple base classes in C++?
   1. Hierarchical Inheritance
   2. Single Inheritance
   3. **Multiple Inheritance**
   4. Multilevel Inheritance
7. What is the keyword used to call the base class constructor from the derived class constructor?
   1. base()
   2. this()
   3. **super()**
   4. parent()
8. In C++, can a derived class access private members of the base class?
   1. Yes
   2. **No**
9. What is the function used to check if a class is derived from another class in C++?
   1. **is\_base\_of()**
   2. derived\_from()
   3. is\_derived\_from()
   4. has\_base()
10. Which type of inheritance involves a class being derived from itself?
    1. Hierarchical Inheritance
    2. Single Inheritance
    3. Multiple Inheritance
    4. **Recursive Inheritance**
11. If a class 'B' is derived from class 'A', and class 'C' is derived from class 'B', which constructor is called first when creating an object of class 'C'?
    1. **Class 'A' constructor**
    2. Class 'B' constructor
    3. Class 'C' constructor
    4. Depends on the order of derivation
12. What is the use of the 'virtual' keyword in inheritance?
    1. It allows private members to be accessed in the derived class
    2. It ensures that only one instance of the base class is created
    3. **It enables late binding and polymorphism**
    4. It prevents the derived class from inheriting certain members
13. In C++, which access specifier is used when deriving a class from a base class to ensure that the public members of the base class remain public in the derived class?
    1. private
    2. protected
    3. **public**
    4. friend
14. What is the syntax to inherit a class 'B' privately from class 'A' in C++?
    1. class B : A;
    2. class B - A;
    3. **class B : private A;**
    4. class B private A;
15. When is the destructor of a base class called when using inheritance in C++?
    1. After the destructor of the derived class
    2. **Before the destructor of the derived class**
    3. It is not automatically called, you have to call it explicitly
    4. At the same time as the destructor of the derived class
16. Which type of inheritance involves creating a chain of classes with each derived class being a base class for the next class in the chain?
    1. Hierarchical Inheritance
    2. Single Inheritance
    3. Multiple Inheritance
    4. **Multilevel Inheritance**
17. In C++, can a derived class have multiple base classes of the same name and type?
    1. Yes
    2. **No**
18. What is the order of constructor and destructor execution in a multiple inheritance scenario in C++?
    1. **Constructor: Base classes in the order of derivation, Destructor: Derived class, Base classes in reverse order of derivation**
    2. Constructor: Derived class, Base classes in reverse order of derivation, Destructor: Base classes in the order of derivation
    3. Constructor: Base classes in reverse order of derivation, Destructor: Base classes in the order of derivation, Derived class
    4. Constructor: Derived class, Destructor: Base classes in the order of derivation, Base classes in reverse order of derivation
19. In C++, which access specifier is used when deriving a class from a base class to ensure that the protected members of the base class remain protected in the derived class?
    1. private
    2. **protected**
    3. public
    4. friend
20. What is the maximum number of levels of inheritance that C++ allows?
    1. 10
    2. 127
    3. 255
    4. **Unlimited**
21. In C++, what is the difference between "protected" and "private" access specifiers when inheriting a class?
    1. Both provide the same level of visibility to the derived class
    2. **"protected" allows access to derived classes, but "private" does not**
    3. "private" allows access to derived classes, but "protected" does not
    4. "protected" members can be accessed in the base class, but "private" members cannot
22. What is hybrid inheritance in C++?
    1. **The combination of single and multiple inheritances**
    2. The combination of public and private inheritance
    3. The combination of hierarchical and multilevel inheritances
    4. The combination of virtual and non-virtual inheritance
23. Can a C++ class be derived from multiple base classes using private inheritance?
    1. **Yes**
    2. No
24. Which type of inheritance involves creating a tree-like structure with a single root class and multiple derived classes?
    1. **Hierarchical Inheritance**
    2. Single Inheritance
    3. Multiple Inheritance
    4. Multilevel Inheritance
25. What is the diamond problem in C++?
    1. A common issue when using virtual functions with multiple inheritance
    2. An issue caused by circular dependencies in header files
    3. A conflict that arises when two derived classes have the same name
    4. **A problem caused by ambiguous access to a member in a multiple inheritance scenario**
26. What is the use of the 'using' keyword in C++ when dealing with inheritance?
    1. **It allows using base class members directly in the derived class**
    2. It specifies which constructor to use in the derived class
    3. It allows renaming derived class members
    4. It enables late binding and virtual function calls
27. In C++, can a derived class override a base class's constructor?
    1. Yes
    2. **No**
28. What is the purpose of the 'protected' access specifier in C++?
    1. To restrict access to members within the same class only
    2. **To make members accessible in derived classes but not in other parts of the program**
    3. To provide full access to all parts of the program
    4. To enable friend classes to access private members
29. What happens when a class is derived privately from a base class in C++?
    1. **All members of the base class become private in the derived class**
    2. All members of the base class become protected in the derived class
    3. All members of the base class become public in the derived class
    4. The base class becomes a friend of the derived class
30. In C++, can a derived class inherit the constructor of its base class?
    1. **Yes**
    2. No