1. Which keyword is used to inherit a class in C++?
   1. class
   2. private
   3. protected
   4. **public**
2. Which type of inheritance allows a class to inherit multiple base classes?
   1. Single inheritance
   2. **Multiple inheritance**
   3. Multilevel inheritance
   4. Hierarchical inheritance
3. Which constructor is called first in the inheritance hierarchy?
   1. **Base class constructor**
   2. Derived class constructor
   3. Both are called simultaneously
   4. None of the above
4. Inheritance in C++ supports the concept of:
   1. Code reusability
   2. Encapsulation
   3. Polymorphism
   4. **All of the above**
5. What is the keyword used to invoke the base class constructor from the derived class constructor?
   1. super
   2. **base**
   3. parent
   4. this
6. Which member function in the base class can be overridden in the derived class?
   1. **Public member functions**
   2. Private member functions
   3. Protected member functions
   4. All member functions can be overridden
7. Which keyword is used to denote that a member function in the derived class is intended to override a base class member function?
   1. **override**
   2. virtual
   3. final
   4. over
8. Which of the following is true regarding function overloading and function overriding in C++?
   1. **Function overloading is resolved at compile-time, while function overriding is resolved at runtime.**
   2. Function overloading is resolved at runtime, while function overriding is resolved at compile-time.
   3. Both function overloading and function overriding are resolved at compile-time.
   4. Both function overloading and function overriding are resolved at runtime.
9. Which destructor is called first in the inheritance hierarchy?
   1. Base class destructor
   2. **Derived class destructor**
   3. Both are called simultaneously
   4. None of the above
10. Can a derived class destructor call the base class destructor explicitly?
    1. Yes
    2. **No**
11. What happens when a derived class object is destroyed?
    1. **Derived class destructor is called first, followed by the base class destructor.**
    2. Base class destructor is called first, followed by the derived class destructor.
    3. Only the derived class destructor is called.
    4. Only the base class destructor is called.
12. Which access specifier is used for inheritance in C++ by default?
    1. public
    2. **private**
    3. protected
    4. It depends on the base class definition.
13. Which of the following is true about constructors in a derived class?
    1. A derived class cannot have its own constructor.
    2. A derived class constructor is always called explicitly from the base class constructor.
    3. A derived class constructor is always called implicitly from the base class constructor.
    4. **A derived class can have its own constructor, which may or may not call the base class constructor.**
14. Which type of inheritance involves a class being derived from its own base class?
    1. Single inheritance
    2. Multiple inheritance
    3. **Multilevel inheritance**
    4. Hierarchical inheritance
15. Can a derived class object access private members of the base class?
    1. Yes
    2. **No**
16. In which order are the constructors executed in a multiple inheritance scenario?
    1. From the most derived class to the base class
    2. **From the base class to the most derived class**
    3. In a random order
    4. The constructors are not executed in multiple inheritance.
17. Can a derived class have more than one constructor?
    1. **Yes**
    2. No
18. Which type of inheritance allows the reuse of code from multiple base classes without forming an is-a relationship?
    1. Single inheritance
    2. **Multiple inheritance**
    3. Multilevel inheritance
    4. Hierarchical inheritance
19. Which of the following is true about virtual functions in C++?
    1. A virtual function cannot be overridden.
    2. A virtual function must have a body in the base class.
    3. **A virtual function can be overridden in a derived class.**
    4. A virtual function can only be declared in the base class and defined in the derived class.
20. Which member function is automatically called when an object is created?
    1. **Constructor**
    2. Destructor
    3. Main function
    4. Overloaded operator function
21. When does the destructor of an object get called?
    1. When the object is created
    2. **When the object goes out of scope**
    3. When the object is assigned a new value
    4. When the object is passed by value to a function
22. What is the purpose of the virtual destructor in C++?
    1. To deallocate memory allocated using the 'new' keyword
    2. To ensure proper destruction of derived class objects
    3. **To avoid memory leaks in polymorphic class hierarchies**
    4. All of the above
23. Can constructors be inherited in C++?
    1. Yes
    2. **No**
24. Which of the following is not a valid access specifier in C++?
    1. public
    2. private
    3. protected
    4. **default**
25. In which order are the destructors executed in a multiple inheritance scenario?
    1. **From the most derived class to the base class**
    2. From the base class to the most derived class
    3. In a random order
    4. The destructors are not executed in multiple inheritance.
26. What is the purpose of a pure virtual function in C++?
    1. To declare an abstract base class
    2. **To force derived classes to implement the function**
    3. To provide a default implementation for a function
    4. None of the above
27. Can a derived class object be assigned to a base class object?
    1. **Yes**
    2. No
28. What happens when a base class pointer is assigned a derived class object?
    1. The assignment is not allowed in C++.
    2. **Only the base class portion of the derived class object can be accessed through the pointer.**
    3. The derived class portion of the object is sliced off, and only the base class portion is accessible through the pointer.
    4. Both the base class and derived class portions of the object can be accessed through the pointer.
29. Which concept allows you to reuse the written code?
    1. Encapsulation
    2. Abstraction
    3. **Inheritance \*\*\*\***
    4. Polymorphism
30. Which of the following explains Polymorphism?
    1. int func(int, int); float func1(float, float);
    2. int func(int); int func(int);
    3. **int func(float); float func(int, int, char);**
    4. int func(); int new\_func();
31. Which of the following shows multiple inheritances?
    1. A->B->C
    2. A->B; A->C
    3. **A,B->C**
    4. B->A