1. What is the purpose of a friend function in C++?
   1. **To access private data members of a class**
   2. To declare an object of a class
   3. To initialize the class constructor
   4. To define global variables
2. Question: Which keyword is used to declare a friend function in a class?
   1. **friend**
   2. this
   3. access
   4. friendfunc
3. Question: A friend function can be called using which object?
   1. Only through objects of the class in which it is declared as a friend
   2. Only through objects of the class in which it is defined
   3. **Through objects of any class in the program**
   4. Through objects of the derived class
4. Question: Can a friend function be a member of the class it is declared as a friend of?
   1. Yes
   2. **No**
5. Question: When is the memory allocated to a friend function in C++?
   1. **Compile-time**
   2. Link-time
   3. Run-time
   4. Friend functions do not have memory allocation
6. Question: Which of the following is a valid syntax to declare a friend function in a class?
   1. **friend void display();**
   2. friend display();
   3. void friend display();
   4. display() friend;
7. Question: Friend functions are used for:
   1. Overloading operators
   2. Inheriting classes
   3. **Accessing private members of a class**
   4. Creating dynamic objects
8. Question: Which class relationship allows one class to access the private members of another class?
   1. Composition
   2. Inheritance
   3. Aggregation
   4. **Friendship**
9. Question: A friend function can be declared in:
   1. Public section of the class
   2. Private section of the class
   3. Protected section of the class
   4. **Any section of the class**
10. Question: Friend classes are used for:
    1. Inheriting private members of another class
    2. **Accessing private members of another class**
    3. Creating objects of another class
    4. Overriding member functions of another class
11. Question: Which access specifier is used for a friend function in a class?
    1. public
    2. private
    3. protected
    4. **No access specifier is required**
12. Question: A friend class can access:
    1. Public members of the class it is a friend of
    2. Private members of the class it is a friend of
    3. Protected members of the class it is a friend of
    4. **All members of the class it is a friend of**
13. Question: In C++, can a friend function access private static members of a class?
    1. **Yes**
    2. No
14. Question: Which of the following statements is true about passing objects to functions in C++?
    1. Objects cannot be passed as function arguments
    2. Objects are always passed by value
    3. Objects are always passed by reference
    4. **Objects can be passed by value or reference**
15. Question: If an object is passed by value to a function, changes made to the object inside the function:
    1. Are reflected in the original object
    2. **Are not reflected in the original object**
    3. Cause a compilation error
    4. Create a copy of the original object
16. Question: When an object is passed by reference to a function, changes made to the object inside the function:
    1. **Are reflected in the original object**
    2. Are not reflected in the original object
    3. Cause a compilation error
    4. Create a copy of the original object
17. Question: Passing an object by reference is more efficient than passing by value because:
    1. It consumes less memory
    2. It prevents changes to the original object
    3. **It avoids the need to copy the entire object**
    4. It allows direct access to private members
18. Question: In C++, the default method of passing objects to functions is:
    1. **Pass by value**
    2. Pass by reference
    3. Pass by pointer
    4. Pass by address
19. Question: To prevent a function from modifying the original object, the parameter should be passed as:
    1. **const**
    2. volatile
    3. mutable
    4. static
20. Question: If a function is defined outside the class definition and needs access to private members, it should be declared as:
    1. private
    2. public
    3. static
    4. **friend**
21. Question: Which of the following is true about the const keyword in C++?
    1. It is used to declare constant variables only
    2. It is used to declare constant functions only
    3. **It is used to make variables immutable**
    4. It is used to make functions immutable
22. Question: In C++, a const member function can be called on:
    1. Const objects only
    2. Non-const objects only
    3. **Both const and non-const objects**
    4. Neither const nor non-const objects
23. Question: In C++, which access specifier is used by default for members of a class defined inside another class?
    1. public
    2. **private**
    3. protected
    4. friend
24. Question: In C++, can a friend function be inherited by the derived class?
    1. Yes
    2. **No**
25. Question: Which of the following is true about friend functions?
    1. They can access only private members of a class
    2. **They can access both private and protected members of a class**
    3. They can access only public members of a class
    4. They cannot access any members of a class
26. Question: What is the scope of a friend function in C++?
    1. Local to the class in which it is declared
    2. Local to the class in which it is defined
    3. **Global to the program**
    4. Global to the file
27. Question: Can a friend function be overloaded in C++?
    1. Yes
    2. **No**
28. Question: A friend function in C++ can be declared in which section of the class?
    1. public
    2. private
    3. protected
    4. **Any section**
29. Question: When a friend function is defined outside the class, how is it identified as a friend of the class?
    1. **By using the keyword friend before the function definition**
    2. By using the keyword friend before the function declaration
    3. By using the keyword friend after the function definition
    4. By using the keyword friend after the function declaration
30. Question: Which of the following is true about friend classes in C++?
    1. **Friend classes can access private members of the class they are friends with**
    2. Friend classes can inherit private members of the class they are friends with
    3. Friend classes can override private members of the class they are friends with
    4. Friend classes can access private members of any class in the program
31. Question: In C++, can a friend class access the protected members of the class it is a friend of?
    1. **Yes**
    2. No
32. Question: How do you declare a friend class in C++?
    1. friend MyClass;
    2. **friend class MyClass;**
    3. friend MyClass::;
    4. friend::MyClass;
33. Question: Can a class be both a friend and a base class of another class in C++?
    1. **Yes**
    2. No