1. What are the six combinations of access modifier keywords and what do they do?
   - Public: member can be accessed anywhere
   - Private: member can only be accessed in current class
   - Protected: member can be accessed in current class and child classes
   - Internal: member can be accessed in the current assembly
   - Protected-internal: member can be accessed in same assembly or in derived class in other assemblies
   - Private-protected: member can be accessed inside containing class or in a class that derives from a containing class, but only in the same project

2. What is the difference between the static, const, and readonly keywords when applied to a type member?
   - Const: be assigned a value at declaration, and this value may not then change at a later time
   - Static: member belongs to the type of an object rather than to an instance of that type
   - Readonly: assignment to that file can only occur as part of the declaration of the class or in a constructor

3. What does a constructor do?
   - A special method which share the same name of the class and doesn't have return type, not even void

4. Why is the partial keyword useful?
   - Indicates that other parts of the class, struct, or interface can be defined in the namespace

5. What is a tuple?
   - Can represent a database record, its components can represent individual fields of the record

6. What does the C# record keyword do?
   - Define a reference type that provides built-in functionality for encapsulating data

7. What does overloading and overriding mean?
   - Overloading: multiple methods in same class, they have same method signature, including access modifiers and method name
   - Overriding: happens between base class and derived class. They have same method signature, including access modifiers, method name, and input parameters

8. What is the difference between a field and a property?
   - Field: a variable of any type that is declared directly in a class
   - Property: a member that provides a flexible mechanism to read, write, or compute the value of a private field

9. How do you make a method parameter optional?
    - Assign default values for that parameter

10. What is an interface and how is it different from abstract class?
    - Interface is a collection methods which are by default abstract and will be implemented by derived classes
    - Abstract class provides a base class to its subclass – use when we have a clear class hierarchy; interface defines common behavior or functionalities that can be implemented by any class – contract

11. What accessibility level are members of an interface?
    - Public default

12. True/False. Polymorphism allows derived classes to provide different implementations of the same method.
13. True/False. The override keyword is used to indicate that a method in a derived class is providing its own implementation of a method.
14. True/False. The new keyword is used to indicate that a method in a derived class is providing its own implementation of a method.
15. True/False. Abstract methods can be used in a normal (non-abstract) class.
16. True/False. Normal (non-abstract) methods can be used in an abstract class.
17. True/False. Derived classes can override methods that were virtual in the base class.
18. True/False. Derived classes can override methods that were abstract in the base class.
19. True/False. In a derived class, you can override a method that was neither virtual non abstract in the base class.
20. True/False. A class that implements an interface does not have to provide an implementation for all of the members of the interface.
21. True/False. A class that implements an interface is allowed to have other members that aren't defined in the interface.
22. True/False. A class can have more than one base class.
23. True/False. A class can implement more than one interface.