1.What are the six combinations of access modifier keywords and what do they do?

- public: can be accessed anywhere
- protected: can be accessed in the current class and child classes
- internal: can be accessed in the current assembly
- private: can only be accessed in the current class
- protected internal: access in current assembly or from derived class in another assembly
- private protected: within declaring assembly or from current class and child classes

2. What is the difference between the static, const, and readonly keywords when applied to a type member?

- Static: member that belongs to the type of object not an instance of the object
- Const: assigned value at declaration that is not change,
- readonly: assignment can only happen at declaration

3. What does a constructor do?

- Create object and initialize its class members

4. Why is the partial keyword useful?

- It allows splitting the definition of an object over multiple files
- Have the benefit of allowing many programmers to work on it at once

5. What is a tuple?

A structure that allows elements of different data types

6. What does the C# record keyword do?

7. What does overloading and overriding mean?

- overloading: compile time polymorphism, when multiple method of the same class have different parameters
- overriding: runtime polymorphism, when child method have the same signature as parent method

8. What is the difference between a field and a property?

- Fields are private to class and is access vial properties

9. How do you make a method parameter optional?

- static public void exampleMethod(int optionalint = 1)

10. What is an interface and how is it different from abstract class?

- Interface is like a class with the same member, but does not contain initialization of members
- Abstract class can have concrete methods interface cannot
- Interface cannot use and initialize its methods, abstract class can

11.What accessibility level are members of an interface?

Public by default

12.True / False.Polymorphism allows derived classes to provide different implementations of the same method.
True

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.
True

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.
True

15. True/False. Abstract methods can be used in a normal (non-abstract) class.
False

16.True / False.Normal(non -abstract) methods can be used in an abstract class.
True

17.True / False.Derived classes can override methods that were virtual in the base class.
True

18.True / False.Derived classes can override methods that were abstract in the base class.
True

19.True / False.In a derived class, you can override a method that was neither virtual non abstract in the base class.
False

20.True / False.A class that implements an interface does not have to provide an implementation for all of the members of the interface.
False

21.True / False.A class that implements an interface is allowed to have other members that aren’t defined in the interface.
True

22.True / False.A class can have more than one base class.
True

23.True / False.A class can implement more than one interface
True