Question 1:

Public: Objects that implement public access modifiers are accessible from everywhere in a project without any restrictions.

Private: Objects that implement private access modifier are accessible only inside a class or a structure. Cannot access them outside the class they are created

Protected: The protected keyword implies that the object is accessible inside the class and in all classes that derive from that class.

Internal: The access is limited exclusively to classes defined within the current project assembly.

Protected Internal: The protected internal access modifier is a combination of protected and internal. As a result, we can access the protected internal member only in the same assembly or in a derived class in other assemblies.

Private Protected: Can access members inside the containing class or in a class that derives from a containing class, but only in the same assembly(project). Therefore, if we try to access it from another assembly, we will get an error.

Question 2:

Constant and ReadOnly keyword is used to make a field constant which value cannot be modified. The static keyword is used to make members static that can be shared by all the class objects. Constant fields must be assigned a value at the time of declaration and after that, they cannot be modified. A readonly field can be initialized either at the time of declaration or within the constructor of the same class. Therefore, readonly fields can be used for run-time constants.

Question 3:

A constructor is a method whose name is the same as the name of its type. Constructors enable the programmer to set default values, limit instantiation, and write code that is flexible and easy to read. It does not include a return type.

Question 4:

1.When working on large projects, spreading a class over separate files enables multiple programmers to work on it at the same time.

2.When working with automatically generated source, code can be added to the class without having to recreate the source file. Visual Studio uses this approach when it creates Windows Forms, Web service wrapper code, and so on. You can create code that uses these classes without having to modify the file created by Visual Studio.

3.Can use source generators to generate additional functionality in a class.

Question 5:

A tuple is a data structure that contains a sequence of elements of different data types. It can be used where you want to have a data structure to hold an object with properties, but you don't want to create a separate type for it.

Question 6:

Record types are immutable reference types that provide value semantics for equality. They are ideal in situations where you are going to need to compare objects and maybe you want to ensure the property values of an object cannot be changed during the execution of other processes.

Question 7:

Overloading occurs when two or more methods in one class have the same method name but different parameters.

Overriding occurs when two methods have the same method name and parameters. One of the methods is in the parent class, and the other is in the child class. Overriding allows a child class to provide the specific implementation of a method that is already present in its parent class.​

Question 8:

A field is a variable of any type that is declared directly in a class. It can be used to explain the characteristics of an object or a class.

A property is a member that provides a flexible mechanism to read, write or compute the value of a private field. It can be used to set and receive values of a field.

Properties expose fields. Fields should (almost always) be kept private to a class and accessed via get and set properties. Properties provide a level of abstraction allowing you to change the fields while not affecting the external way they are accessed by the things that use your class.

Question 9:

There are four ways:

Use Parameter arrays with keyword “params”.

Default parameter by setting parameter to a default value in the method definition.

OptionalAttribute with keyword “[Optional]”.

Method Overloading: provide several overloading methods for possible optional parameter

Question 10:

An abstract class allows you to create functionality that subclasses can implement or override. An interface only allows you to define functionality, not implement it. And whereas a class can extend only one abstract class, it can take advantage of multiple interfaces.

Question 11:

Interface members are public by default because the purpose of an interface is to enable other types to access a class or struct.

Question 12:

True

Question 13:

True

Question 14:

True

Question 15:

False

Question 16:

True

Question 17:

True

Question 18:

True

Question 19:

False

Question 20:

False

Question 21:

True

Question 22:

False

Question 23:

True