Assignment 3

■ OOP

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

Public: members can be accessed anywhere
Protected: members can be accessed in current class and children classes
Internal: member can be assessed in current assembly
Private: members can be accessed only in current class
Protected internal: combination of protected and internal.
Private protected: combination of private and protected keywords.

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

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.

3. What does a constructor do?

Constructor is a special method in the class which shares the same name with class but don't have any return type. It's used to create the instance of a class.

4. Why is the partial keyword useful?

Partial keyword is useful because it can have multiple programmers work together or it can add some features to a class without modifying the source code.

5. What is a tuple?

Tuple is a data structure in C# which may consists of multiple parts of data.

6. What does the C# record keyword do?

Record keyword can be used to define a reference type that provides built-in functionality for encapsulating data.

7. What does overloading and overriding mean?

Overloading means rewriting methods in the same class sharing the same name, but having different input and output. Overriding means rewriting methods in base class and its subclasses which shares the same method name and same input parameters.

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

In C#, a field is a variable (that can be of any type) that is defined inside a class. It can be used to define the characteristics of an object or a class. On the other hand, a property is a member of the class that provides an abstraction to set (write) and get (read) the value of a private field.

9. How do you make a method parameter optional?

There are 2 ways. First is to assign a default value to a parameters, another one is to use method overloading.

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

Interface is a collection of methods which are by default abstract and public, and will be implemented by the derived classes. It's must be implemented, which means it can not be inherited. Methods in Interface are by default abstract and public while methods in abstract class can be non-abstract.

11. What accessibility level are members of an interface?

Public.

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.

False

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.
False

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