Object oriented programming

* 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
* Protected-The protected keyword implies that the object is accessible inside the class and in all classes that derive from that class
* Internal - For Internal keyword, 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
* Private protected-The private protected access modifier is a combination of the private and protected keywords.
* Static - Static members can only be accessed within the static methods. The non-static methods cannot access static members.
* Read-only- Read-only fields can be initialized at declaration or in the constructor. Therefore, read only variables are used for the run-time constants.
* Const- The constant fields must be initialized at the time of declaration. Therefore, const variables are used for compile-time constants.

1. A constructor allocates memory and initializes field values.
2. You can use the partial keyword to split the definition of a type of over multiple files.
3. A data structure consisting of multiple parts.
4. define a reference type that provides built-in functionality for encapsulating data.
5. Overloading is when you define more than one method with the same method name and different input parameters. Overriding means same method name and same parameter,

occur in different class that has inheritance relationship.

1. A field is a data storage location that can be referenced. A property is one or a pair of methods that get and/or set a value. The value is often stored in a private field.
2. You make a method parameter optional by assigning a default value to it.
3. 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.
4. Public or internal
5. True
6. True
7. False
8. False
9. False
10. True
11. True
12. True
13. false
14. False
15. False
16. True