

1. Access Modifier:
 - a) Public: Access from anywhere in program
 - b) Private: Access from within class defined
 - c) Protected: Access from within class defined and inherited classes
 - d) Internal: Within same assembly
 - e) Protected Internal: Protected + Internal
 - f) Private Protected: similar to Protected but limited to derived classes in same assembly.
2. Static: belongs to the class/method/type, const is a kind of variable that its value cant be changed after initialization, read-only is a constant only be assigned during declaration.
3. Constructor is called when a new object of its class is created, doing initialization for the newly created object.
4. Partial is used to allow code to be separated across multiple files, for better organization.
5. Tuple allows temp storage of some items of different kind, without the need of create a new class.
6. It's a builtin data type that are some degree similar to class but having some additional functions.
7. Overloading: same method name in a class but with different input parameter. Overriding: same method name in a child class and in its base class, child class would overload the method in the base class.
8. Property hides fields through abstraction, fields are individual variables, and property usually provides gets and set method for the fields.
9. To create another method without the needed parameter using overloading or provide a default value for the parameter
10. Abstract class will provide base class to its subclasses; is a wise choice when we have clear hierarchy relationship. Interface will define common behaviors and functionalities that can be implemented by any class. Once class can only inherit from one abstract or concrete class but one class can implement multiple interfaces. Methods in abstract class can be abstract method or non-abstract method. But methods in an interface are public and abstract by default.
11. Public
12. True
13. True
14. False
15. False
16. True
17. True
18. True
19. False
20. False
21. True
22. False
23. True