



Conceptual Questions



By: Amanuel Legesse

ID: ATE/3628/11

Submitted To: kabila Haile

Conceptual Questions

1. What is an Object?

- ✓ In OOP an object is something we think about first while designing a program, An object is a component of a program that knows how to perform certain actions and how to interact with other elements of the program.

2. What is Encapsulation?

- ✓ Refers to the bundling of data, along with the methods that operate on that data, into a single unit.

3. What is Abstraction?

- ✓ Abstraction is used to hide background details or any unnecessary implementation about the data so that users only see the required information. It is one of the most important and essential features of object-oriented programming.

4. Which are Access Specifiers?

- ✓ C# has 5 access specifier; those are private, public, internal, protected and protected Internal.

5. What is Inheritance?

- ✓ Inheritance is a mechanism of acquiring the features and behaviors of a class by another class

6. How can you implement multiple inheritance in C#?

- ✓ C# does not support multiple class inheritance. To overcome this problem we use interfaces to achieve multiple class inheritance. With the help of interface a class can have access to another class.

7. Are private class members inherited to the derived class?

- ✓ private members of the base-class are not accessible within the derived-class and to the instances of derived-class.

8. What is Polymorphism?

- ✓ In c#, Polymorphism means providing an ability to take more than one form, polymorphism provides an ability for the classes to implement different methods called through the same name. It also provides an ability to invoke a derived class's methods through base class reference during runtime based on our requirements.

9. What is method Overloading?

- ✓ Method overloading is a concept of in which we can create multiple methods of the same name in the same class, and all methods work in different ways

10. When and why to use method Overloading?

- ✓ Overloading occurs when two or more methods in one class have the same method name but different parameters.
- ✓ so that they can call the same method for different types of data. This makes the code look clean. This reduces the execution time because the binding is done in compilation time itself.

11. What is method Overriding?

- Method Overriding is a technique that allows the invoking of functions from another class (base class) in the derived class.

12. What is Constructor?

- A constructor is a special type of member function that is called automatically when an object is created.

13. Describe some of the key points regarding the Constructor.

- An interface cannot have the constructor.
- A constructor cannot be abstract, static or final
- A constructor can be overloaded.
- Constructors cannot return a value.
- Constructors do not have a return type; not even void.
- An abstract class can have the constructor.
- Constructors are automatically called when an object is created.

14. What is Private Constructor?

- A private constructor is a special instance constructor. It is generally used in classes that contain static members only.

15. Can you create object of class with private constructor in C#?

- No, object of a class having private constructor cannot be instantiated from outside of the class.]

16. What is the use of private constructor in C#?

- Private constructors are used to prevent creating instances of a class when there are no instance fields or methods,

17. What is the use of static constructor in C#?

- A static constructor is used to initialize any static data, or to perform a particular action that needs to be performed only once. It is called automatically before the first instance is created or any static members are referenced.

18. What is Destructor?

- Destructors are methods inside the class used to destroy instances of that class when they are no longer needed.

19. What is Namespaces?

- Namespaces are used to organize and provide a level of separation of codes.
- It is a declarative region that provides a scope to the identifiers (the names of types, functions, variables) inside it.

20. What are Virtual, Override, and New keywords in C#?

- **The Virtual keyword** is used for generating a virtual path for its derived classes on implementing method overriding. The Virtual keyword is used within a set with an override keyword.
- **The Override keyword** is used in the derived class of the base class in order to override the base class method. The Override keyword is used with the virtual keyword.
- **The New keyword** is also used for polymorphism but in the case of method overriding. We can say that we are changing what the base class does for the derived class.

21. What is the difference between Struct and Class in C#?

- Struct are value types whereas Classes are reference types
- Structs can be instantiated without using a new operator. A struct cannot inherit from another struct or class, and it cannot be the base of a class.

- In classes, two variables can contain the reference of the same object and any operation on one variable can affect another variable
- In structs, each variable contains its own copy of the data and an operation on one variable does not affect another variable.

22. What is Interface?

- An interface is a specification for a set of class members, not an implementation. An Interface is a reference type and it contains only abstract members.

23. Why to use Interfaces in C#?

- The Interface in C# is a fully un-implemented class used for declaring a set of operations of an object. So, we can define an interface as a pure abstract class which allows us to define only abstract methods.

24. What is Implicit interface implementation?

- An implementation of an interface is a program that references the interface using the implements keyword. The program is required to provide method logic for all non-default methods. Optionally, the program can provide an implementation of a default method defined in the interface.

25. What is Explicit interface implementation?

- With explicit implementations, in the class the interface members are not declared as public members and cannot be directly accessed using an instance of the class, but a cast to the interface allows accessing the members.

26. What is Abstract class?

- An abstract class is a class that is declared abstract. It may or may not include abstract methods. Abstract classes cannot be instantiated, but they can be subclassed.

27. Describe Abstract class in detail.

- A class that is declared using “abstract” keyword is known as abstract class. It can have abstract methods as well as concrete methods. A normal class(non-abstract class) cannot have abstract methods.

28. What is the difference between Abstraction and Encapsulation?

- Abstraction allows us to represent complex real world in simplest manner. It is process of identifying the relevant qualities and behaviors an object should possess, in other word represent the necessary feature without representing the back ground details.
- Encapsulation, it is a process of hiding all the internal details of an object from the outside real world. The word Encapsulation, like Enclosing into the capsule. It restrict client from seeing its internal view where behavior of the abstraction is implemented.
- Abstraction is the process or method of gaining the information. While encapsulation is the process or method to contain the information.

29. Can Abstract class be Sealed in C#?

- An abstract class can contain sealed methods. The abstract method or class cannot be declared as sealed. A subclass of an abstract class can only be instantiated if it implements all of the abstract methods of its superclass.

30. Can abstract class have Constructors in C#?

- Yes, an abstract class can have a constructor. In general, a class constructor is used to initialize fields. Along the same lines, an abstract class constructor is used to initialize fields of the abstract class.

31. Can you declare abstract methods as private in C#?

- If a method of a class is private, we cannot access it outside the current class, not even from the child classes of it. But, in case of an abstract method, we cannot use it from the same class, we need to override it from subclass and use. Therefore, the abstract method cannot be private.

32. Can abstract class have static methods in C#?

- Yes, abstract class can have Static Methods. The reason for this is Static methods do not work on the instance of the class, they are directly associated with the class itself.

33. Does Abstract class support multiple Inheritance?

- No, C# doesn't support multiple inheritance.

34. Abstract class must have only abstract methods. Is it true or false?

- It is false, we can declare an abstract class with no abstract methods.

35. When do you use Abstract Class?

- An abstract class is used if we want to provide a common, implemented functionality among all the implementations of the component.
- An abstract class is a great choice if we are bringing into account the inheritance concept.

36. Why can Abstract class not be Instantiated?

- We cannot instantiate an abstract class because it is abstract.

37. Which type of members can you define in an Abstract class?

- we can add any type of members in abstract class. In general, the data members of a class should be initialized and assigned to only within the constructor and other member functions of that class.

38. What is Operator Overloading?

- Operator overloading gives the ability to use the same operator to do various operations. It provides additional capabilities to C# operators when they are applied to user-defined data types.

39. Is it possible to restrict object creation in C#?

- We can limit the number of object creation of class in C# using the static variable. Static variable is used to share the value to all instance of that class.

40. Can you inherit Enum in C#?

- No. It is not possible. Enum cannot inherit in derived class because by default Enum is sealed.

41. Is it possible to achieve Method extension using Interface?

- We can use extension methods to extend a class or interface, but not to override them.

42. Is it possible that a Method can return multiple values at a time?

- We can return more than one values from a function by using the method called “call by address”, or “call by reference”.

43. What is Constant?

- Constants are immutable values which are known at compile time and do not change for the life of the program.

44. What is Readonly?

- The readonly keyword is a modifier that indicates that assignment to the field can only occur as part of the declaration or in a constructor in the same class.

45. What is Static?

- Static modifier declares a static member, which belongs to the type itself rather than to a specific object. The static modifier can be used to declare static classes

46. What is Static ReadOnly?

- A Static Readonly type variable's value can be assigned at runtime or assigned at compile time and changed at runtime.