**Object Oriented Programming Langage Part 4**

**33. Can you allow class to be inherited, but prevent the method from being over-ridden?**

Yes, just leave the class public and make the method *sealed*.

**34. Will sealed class allows inheritance, if not why?**

Sealed means it is not inheritable

**35. What are the advantages of Private constructor?**

1. Private constructor will prevent the user from creating the instance of the class which contains only static members.
2. Private constructor are used for implementing the singleton pattern

**36. While using inheritance, derived class construct will call base class constructor?**

Yes, base class constructor will be called before child class constructor

**37. Overloaded constructor will call default constructor internally?**

No, overload constructor will not call default constructor

**38. What is difference between Overrides and Overridable?**

*Overridable (VB.Net)/ virtual (C#)* is used in parent class to indicate that a method can be overridden.*Overrides(VB.Net)/ override(C#)* is used in the child class to indicate that you are overriding a method.

**39. What is Method overloading?**

Method overloading occurs when a class contains two methods with the same name, but different signatures.

**40. What is operator overloading?**

Operator overloading is used to provide a custom functionality to existing operators. For Example +,-,\* and / operators are used for mathematical functionality. But we can overload these operators to perform custom operation on classes or structure.

Example:

To concatenate the two strings we have to use *Concat* method

Dim str1, str2, str3 As String

str1 = "Hello"

str2 = "world"

str3 = String.Concat(str1, str2)

But .Net provides in build operator overloading for string we can use ‘+’ operator for concatenating the string value

str3=str1+str2

Similarly we can also implement operator overloading for classes or structure

Employee3= Employee1 + Employee2

**41. What is mean by abstraction?**

Abstraction is the process of showing necessary information and hiding unwanted information. Let us consider the "CalculateSalary" in your Employee class, which takes EmployeeId as parameter and returns the salary of the employee for the current month as an integer value. Now if someone wants to use that method. He does not need to care about how Employee object calculates the salary? An only thing he needs to be concern is name of the method, its input parameters and format of resulting member

**42. What is mean by abstraction class?**

Abstract classes contain one or more abstract methods that do not have implementation. An abstract class is a parent class that allows inheritance but can never be instantiated. Abstract classes allow specialization of inherited classes.

**43. What id mean by Interface?**

Interface defines the set of properties, signature of the methods and events. It does not include any implementation. Class which implements the interface can provide the behavior to the implemented method. For example two class*MyEnglishClass* and *MyFreanchClass* implementing same interface and provide two different set of behavior in their implementation.

public interface IMyInterface

{

string Hello(string name);

}

public class MyEnglishClass:IMyInterface

{

public string Hello(string name)

{

return "Hello " + name;

}

}

public class MyFrenchClass : IMyInterface

{

public String Hello(string name)

{

return "allo " + name;

}

}

**44. What is difference between Abstract class and Interface?**

* In Interface all the method must be abstract; in abstract class we can have both abstract and concrete methods.
* Access modifiers cannot be specified in interface because it should always be public; in Abstract class, we can specify the access modifier.

**45. In which Scenario you will go for Abstract or Interface Class?**

Abstract classes are useful when creating components because they allow you specify an invariant level of functionality in some methods, but leave the implementation of other methods until a specific implementation of that class is needed. They also version well, because if additional functionality is needed in derived classes, it can be added to the base class without breaking code.

Interfaces are often used to describe the peripheral abilities of a class, not its central identity, e.g. an Automobile class might implement the Recyclable interface, which could apply to many otherwise totally unrelated objects.

**46. What is mean by polymorphism?**

Polymorphism means the ability to take more than one form. An operation may exhibit different behaviors in different instances. The behavior depends on the data types used in the operation. Polymorphism is extensively used in implementing Inheritance.