**C# Basic Questions**

1. **String and string builder - When to use which?**

String Builder refers to mutable string as if the value of string can be changed at later stage after initialization. While String is immutable which means that no modification can be done in existing string object.

So, if string is going to be constant during entire program, then use string class and if string can be changed during program use StringBuilder class.

1. **What is the difference between while and do-while? When to use which?**

While loop is only executed if the given condition in the brackets is true. However, do-while loop will execute first and then check the condition.

So, if you want your program to run at-least once irrespective of the condition given, then use do-while loop. And, if you want your program to run only if the condition is true then use while loop.

1. **What is ref and out keyword? When to use which? with example.**

The ref and out keywords are used in c# for passing the arguments as a reference. They both are treated as same at compile time but at run time they are treated differently.

While passing variable as ref, it is necessary to initialize them otherwise it will give error. However, in case of out, there is no requirement to initialize parameter.

In called method, it is not required to initialize parameter passed as ref. But, if parameter is passed as out, we are required to initialize the parameter inside called method.

We use out generally when we want the called method to initialize values to the variables.

Class abc{

Public static void Main(){

Int x;

Int y;

Int sum;

Sum(out x, out y, out sum);

Console.WriteLine(“the sum of two numbers is {0}”, sum);

}

Public void sum(out x, out y, out sum)

{

x = 50;

y = 70;

sum = x + y;

}

}

We use ref when we want to send values and want the method to take actions

on them.

Class abc{

Public static void Main(){

Int x=50;

Int y=70;

Int sum;

Sum(ref x, ref y, ref sum);

Console.WriteLine(“the sum of two numbers is {0}”, sum);

}

Public void sum(ref x, ref y, ref sum)

{

sum = x + y;

}

}

1. **What is the use of Virtual, override and new keyword? Explain with example.**

**Virtual Keyword**  
Virtual keyword is used to create a virtual path for its derived classes when we implement method overriding.

Class parent

{

Public virtual void display()

{

Console.WriteLine(“This is parent class”);

}

}

**Override Command**

Override command is used to override methods of the parent class in a child class

Class parent

{

Public virtual void display()

{

Console.WriteLine(“This is parent class”);

}

}

Class child : parent

{

Public override void display()

{

Console.WriteLine(“This is child class”);

}

}

Thus, when display method is called it will refer to display method of child class.

**New keyword**

Using new keyword, we can change base class method with derived class method.

Class parent

{

Public void display()

{

Console.WriteLine(“This is parent class”);

}

}

Class child : parent

{

Public new void display()

{

Console.WriteLine(“This is child class”);

}

}

Display method in child class will hide display method of parent class.

Thus, when child obj = new child(); calls display method obj.display(), answer would be “This is child class”.

1. **How to implement readonly property So that I can not assign value to property outside of class but in class's constructor I can assign value to that property**

If we make fields as readonly, then assignment of such fields is only part of declaration or in a constructor of a same class.

Class abc{

Public readonly string s;

Public readonly string s2 = “Sheth”;

Public abc(string s1)

{

s = s1;

Console.Write(“String is : {0} {1}”, s, s2);

}

}

So, here we can assign value to string s inside the constructer of class abc.

**6) Can we define variables in interface?**

**Ans.** No, we can not define variables in interface.