**INTRODUCTION TO C# EXERCISE -I**

**Q1.** Write a program to show the difference between Static & Instance Member?(P)

**SOLUTION:- In git repository**

**Q2.** Write a program to find the leap year? (P)

**SOLUTION:-In git repository**

**Q3.** Write a program To find Constant & vowel using both Switch case & if\_else?(P)

**SOLUTION:-In git repository**

**Q4.** Write a program to define the boxing and unboxing concepts?(P)

**SOLUTION:-In git repository**

**Q5.** Difference between Namespace and assemblies ? (T)

|  |  |
| --- | --- |
| **Namespace** | **Assembly** |
| Name decided by developer at design time | Assembly is used to decide the scope of a type at run time |
| It is basically collection of classes | Assembly can code of the form MSIL |
| Classes available in your program will be logically grouped together under a namespace. | Logical units are physically grouped together as assembly. |
| It can include multiple assemblies | An assembly can contain types belonging to different namespaces. |
| Namespace doesn't have any classification. | Assembly can be classified as private assembly and public assembly. |
| Namespaces have to be mentioned in Project-Properties. | Assemblies need not be explicitly specified. They are automatically described in metadata and manifest files. |
| Namespaces can be nested. | nesting is not permissible in assemblies. |

**Q6.** Write a program to define the scope of different access modifiers?(P)

**SOLUTION:-in git repository**

**Q7.** Write a program to reverse your full name without using an inbuilt function?(P)

**SOLUTION:-In git repository**

**Q.8**.Write a program in C# Sharp to display the multiplication table vertically from 1 to 5 (P)

**SOLUTION:-In git repository**

**Q.9**.What is the value data type and reference data type in c# ? (T)

**SOLUTION:-**

**Value data type :-**

1. Allocated on stack

2. variable contain the value itself.

3. When copy, actual data is copied and each variable can be independently manipulated

4.Integer,float , double, boolean are value type

5. Struct is value type

6.Variables that are based on value types directly contain values.

6.All value types are derived implicitly from the **System.ValueType**

7. A value type cannot contain null value.

8.Each value type has an implicit default constructor that initializes the default value of that type.

int a = new int(); same as int a=0;

9.All of the simple types are aliases of the .NET Framework System types. For example, int is an alias of System.Int32

int x = 123;

System.Int32 x = 123;

**Reference Data Type:-**

1. allocated in heap

2. variable contain the address of the memory location where data is actually stored.

3. only memory address is copied

4. string and object are reference type

5. Classes and interface are reference type