|  |
| --- |
| **DAY 20 ASSIGNMENT**  **By**  **ARUN KUMAR YADLAPALLI**  **@**  **NB Healthcare Technologies PVT LTD.** |

|  |
| --- |
| **Q1) Scope of variables in C#** |
| **A)** The part of a program where the particular variable is accessible is termed as the scope of the variable.  A variable can be defined in a class, method, loop etc..  Three categories of Scope of Variables are   * Class level scope: Declaring any variable in a class but outside any method can be directly accessed anywhere in the class. * Method level scope : Variables that are declared inside method have method level scope. * Block level scope : These variables are generally declared inside the for,while,if, switch etcc... |

|  |
| --- |
| **Q2) What are delegates in C# Write the points discussed about delegates in the class Write C# code to illustrate the usage of delegates.** |
| **Points about Delegates:**   * Delegate is like a function pointer. * Using Delegates, we can call or point to one or more methods. * When we declare a delegate, returntype and parameters must be same as method you want to point using delegate. * Benefit of using delegate is , using single call from delegate all your methods pointing to delegate will be called.   **Types of Delegates:**   1. Single cast Delegate: It points to only one method. 2. Multi-cast Delegate: It points to more than one method. |
| **Program:**  namespace Delegates\_Program  {  //Author: Arun  //Purpose: Using Delegates in the program    class Program  {  public delegate void Arun(int a, int b);    public static void Add(int a, int b)  {  Console.WriteLine(a + b);  }  public static void Mul(int a, int b)  {  Console.WriteLine(a \* b);  }  static void Main(string[] args)  {  Console.WriteLine("Adding and Multiplication");  Arun a = new Arun(Add);  a += Mul;  a(6, 7);  Console.WriteLine("Addition removal");  a -= Add;  a(3, 6);  Console.ReadLine();  }  }  } |

|  |
| --- |
| **Q3) What are nullable types in C# WACP to illustrate nullable types Write some properties of nullable types (like HasValue)** |
| **Code:**  namespace Nullable\_Types  {  //Author: Arun  //Purpose: Nullable types  class Program  {  static void Main(string[] args)  {  int? i = null;    if (i.HasValue)  Console.WriteLine(i);  else  Console.WriteLine("No value");  Console.ReadLine();    }  }  } |
| **Output:** |

|  |
| --- |
| **Q4) out, ref - parameters please research on these two types of parameters write a C# program to illustrate the same.** |
| **Code:**  **Call by reference:**  namespace CallByReference  {  class Program  {  public static void Add(ref int b)  {    b= b\*b;  Console.WriteLine("inside method:" + b);  }    static void Main(string[] args)  {  int a = 5;  Console.WriteLine("Before value" + a);  Add(ref a);  Console.WriteLine("After value" + a);  Console.ReadLine();    }  }  }  **Call by Value:**  namespace CallByValue  {  class Program  {  public static void Add(out int p, out int q)  {  p = 30;  q = 4;    }  static void Main(string[] args)  {  int i, j;  Add(out i, out j);  Console.WriteLine(i);  Console.WriteLine(j);    Console.ReadLine();      }  }  } |
| **Output:**  **Call by Reference:**    **Call by Value:** |