ASSIGNMENT

1) Write a small program where you need to implement a Try and Catch Block .

Code:

```
using System;

public class Program
{
        public static void Main()
        {
                string str = null;

         try
         {
            Console.WriteLine(str[0]);
         }

         catch(NullReferenceException e)
         {
            Console.WriteLine(e.Message);
         }



        }
}
```

Output:

```
Object reference not set to an instance of an object.
```

2) When should we write multiple catch blocks for a Single Try block?

=> We write multiple catch blocks for a Single Try block when we want to handle different type of exceptions in different ways. Each catch block can be used to catch a specific type of exception, allowing you to handle each type of exception differently.

Example:

```
class Program
    {
        static void Main(string[] args)
        {
            int[] arr = { 1, 2, 0, 3 };

            try

            {
                Console.WriteLine(arr[8]);
                int a = 2;
                int b = 0;
                int c = a / b;

                Console.WriteLine(c);



            }


            catch (IndexOutOfRangeException e)
            {
                Console.WriteLine(e.Message);
            }
            catch (DivideByZeroException e)
            {
                Console.WriteLine(e.Message);

            }
              Console.ReadKey();
            }
    }
```

3) How to define a delegate and call any method or event using it?

Code:

```
class DelegateClass
   {
        public  delegate void addnum(int a, int b);
        public delegate void subnum(int a, int b);
        public delegate void str(string s);



        public void sum(int a,int b)
        {
            Console.WriteLine("The sum is :{0} ", a + b);

        }

        public void subtract(int a ,int b)
        {
            Console.WriteLine("The difference between no is : {0} ", a - b);

        }
        public void String(string s)
        {
            Console.WriteLine("My name is {0}", s);
        }


        static void Main(string[] args)
        {
            DelegateClass obj = new DelegateClass();
            addnum d1 = new addnum(obj.sum);
            subnum d2 = new subnum(obj.subtract);
            d1(50,50);
            d2(100, 20);

            str d3 = new str(obj.String);
            d3("Ajay Bohra");



            Console.ReadKey();


        }
   }
```

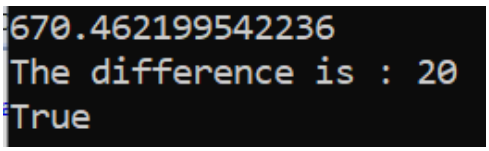
Output:

```
The sum is : 100
The difference between no is : 80
My name is Ajay Bohra
```

4) Try to use Func, Action and Predicate any program.

```
class Program
    {
        static void Main(string[] args)
        {
            Func<int, float, double, double> obj1 = new Func<int, float, double,
double>(AddNumber);
            double sum = obj1.Invoke(400,32.23f,238.2322);


            Console.WriteLine(sum);


            Action<int, int> obj2 = new Action<int,int>(SubNumber);
            obj2.Invoke(100,80);


            Predicate<string> obj3 = new Predicate<string>(CheckLength);
            bool str = obj3.Invoke("Ajay Bohra");
            Console.WriteLine(str);
            Console.ReadKey();
        }
        public static double AddNumber(int no1, float no2, double no3)
        {
            return no1 + no2 + no3;
        }

        public static void SubNumber(int x, int y)
        {
            Console.WriteLine("The difference is : {0} ",x-y);
        }
        public static bool CheckLength(string name)
        {
            if (name.Length > 3)
                return true;
            return false;
        }
    }
```

Output:



5) What will be the output of below code snipped :

```
static void Main()
{
 Func <string,string> output=delegate(string name)
{
 return "Hello" + name;
};
Console.Write(output("James"));
}

static void Main()
{
 Action <int> output = i=>Console.Write(i);
 output(10);
}
```

Output: Hello James
10

6) Write a program to implement Async await with proper justification.

```
class Program
   {
       static void Main(string[] args)
       {
           Console.WriteLine("Main method start.");
           Add();
           Console.WriteLine("Main method end. ");
           Console.ReadKey();
       }

       public async static void Add()
       {
           int x;
           int y;

           Console.WriteLine("Enter the value of x: ");
           x =int.Parse( Console.ReadLine());
           Console.WriteLine("Enter the value of x: ");
           y = int.Parse(Console.ReadLine());

           Console.WriteLine("Adding Number");
           await Task.Delay(1000);
           Console.WriteLine("The sum is : {0} ",x+y);
       }
   }
```

Output:

```
Main method start.
Enter the value of x:
43
Enter the value of x:
32
Adding Number
Main method end.
The sum is : 75
```