4.1

In a written report, critically review and test a .Net programming solution

- Consider 3.2 as example and prepare a report with annotated screenshots of the test.
- Review and test the code in terms of debugging and naming conventions, if not proper change according to the naming convention.

Debugging-:

It is the process of finding the error in the code and solving the code in the code which prevents correct operations of system and it is used in many software companies for testing the code and log flies etc. and the tools which are observed in the visual studio and debugging tools is used to help the developer at the time of development.

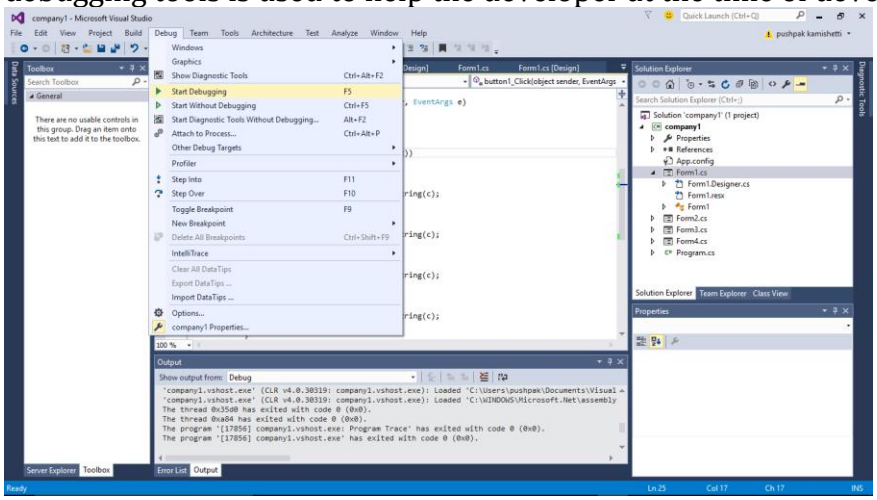

Figure 1selecting the debug

After selecting the start debugging, will check if there are any errors in the code and execute the errors if there, if not will show the output and I can see the debug on the top after that by clicking to start debugging this process will go on.

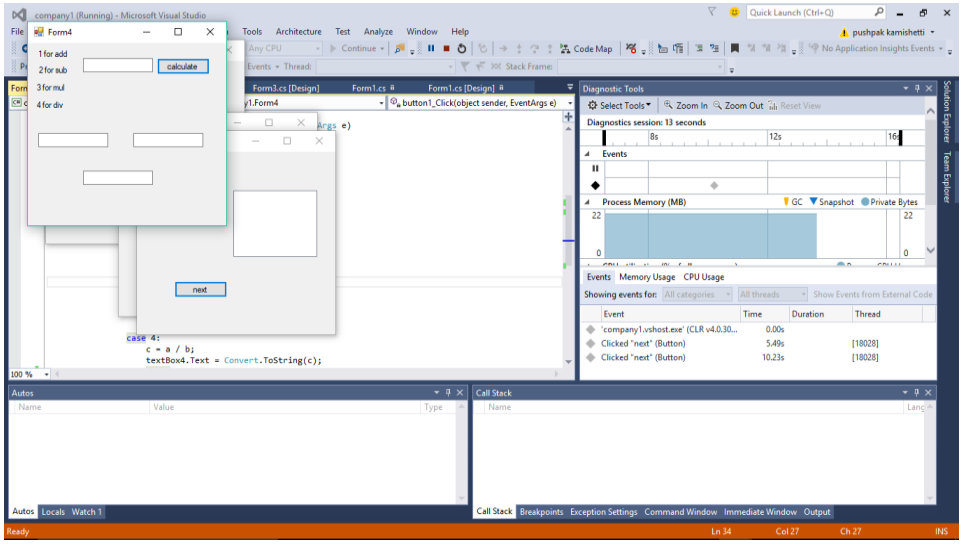

Figure 2output of the code

This is the output of the coding after debugging the code and it doesn't show any problem at the time of execution

Name conventions-:

This is the set of rules given to the program, is used in the code and it is applied at the time of creating the text scripts for programming. These have many different purposes like adding to scripts and functionality in particular language. These codes have capitalization, punctuation to ad symbols and identifiers for few functions.

Camel notations-:

This notation will indicates that first letter is non-capital but first letter of second part should be capital and starting statement of the code may or may not be capital. For example userSystem

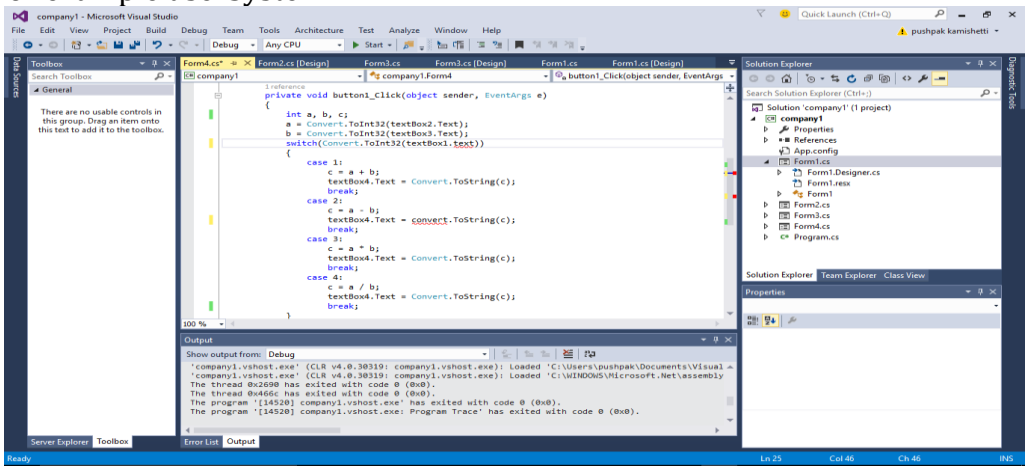

Figure 3error

The above screenshot indicates about the error in the coding because it should use camel notations and red line under should be used for convention.

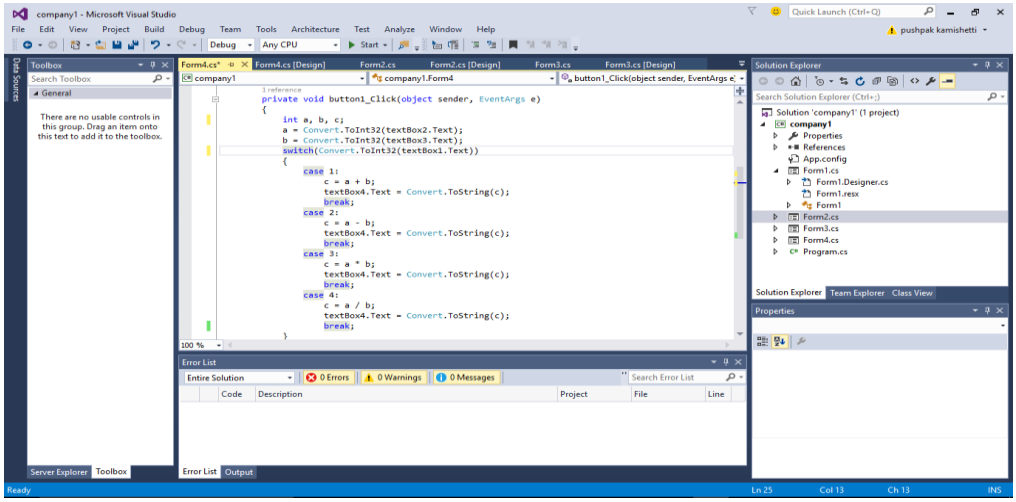

Figure 4output code

The code after clearing the error in the code by using the naming conventions, it got cleared by using camel notation.

Pascal notation-:

This notation will indicates that first letter is capital but first letter second part should be capital and it is manly used for the classes, methods and properties, first letter in identifier and subsequent should be capital. For example WriteLine

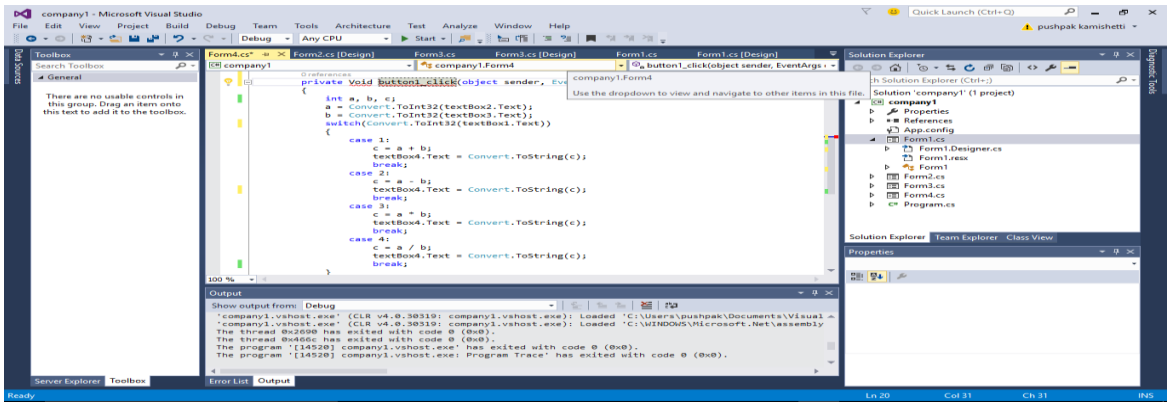

Figure 5error

In the above screenshot I have seen that error may observe in the code and by using Pascal notation, I have cleared the error in the name and error which is got is seen underline of red

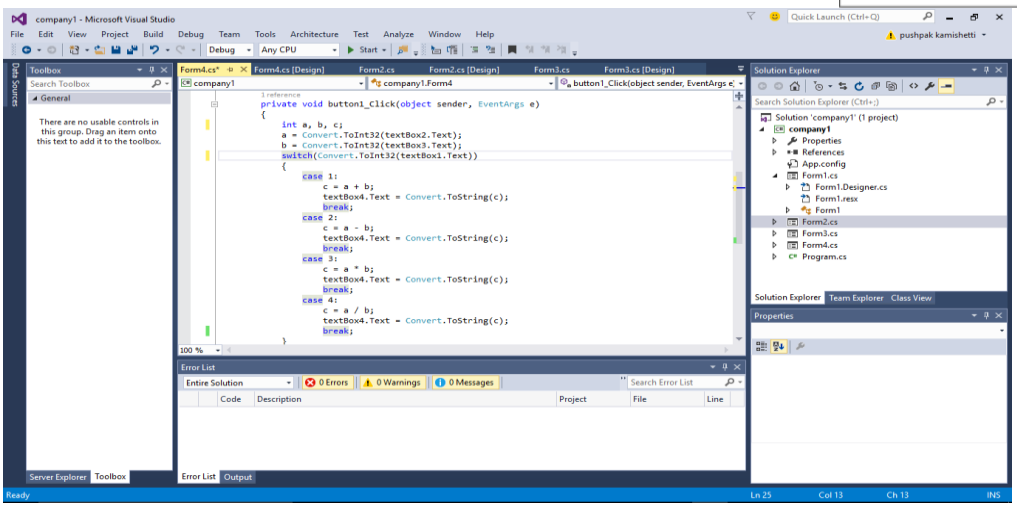

Figure 6output code

The code after clearing the error in the code by using the naming conventions, it got cleared by using Pascal notation

Punctuation-:

The code should include correct punctuation like at the place of '.' It should be and at the place ';' it should and etc. punctuation should be right.

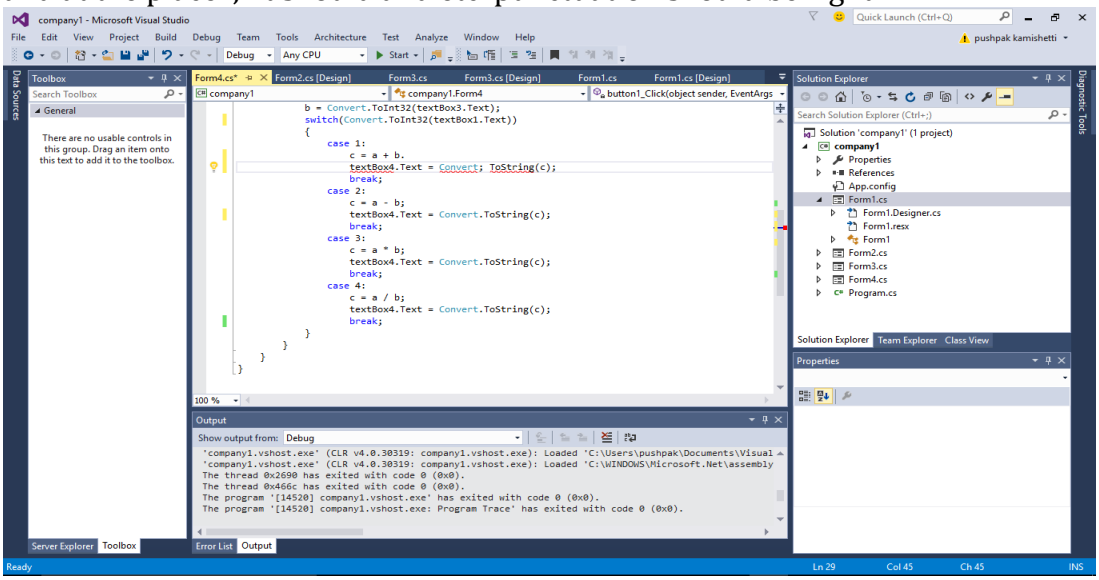

Figure 7error in code

In the above screenshot I have seen that error may observe in the code and by using correct punctuations, I have cleared the error in the name and error which is got is seen underline of red

4.3

Evaluate independent feedback on a developed .Net program solution and make recommendations for improvements, in a written report

- Consider a .Net program 4.2
- Record feedback on conducting surveys, questionnaire
- Analyze the feedback and provide suggestions for improvement

The .net program I have used is finding prime number

Code-:

```
using System;                          //using name space system
namespace pushpak                      //using alternate namespace as pushpak
{
   class one                           //creating class as one
   {
      public static void Main()        //methods starts form there
      {
         Console.Write("Enter a Number : ");          //indicating this sentence to appear on
console screen
         int number;                   //class member
         number = Convert.ToInt32(Console.ReadLine());       //converting it to number
         int x;
         x = 0;
         for (int i = 1; i <= number; i++)
         {
            if (number % i == 0)
            {
               x++;
            }
         }
         if (x == 2)
         {
            Console.WriteLine("Number is a Prime Number and the Largest Factor is
{0}",number);
         }
         else
         {
            Console.WriteLine("Not a Prime Number");
         }
         Console.ReadLine();
      }
   }
}
```

Questionnaire-:

Name-:

Registration number-:

1. What are the strength and weakness of the coding?
2. What are improvements that can be done in this coding?
3. Is the coding is understandable to you and easy to learn?
4. Is the code is working properly?
5. What is the function of this code?
6. Is every naming conversion are used correctly in this coding?

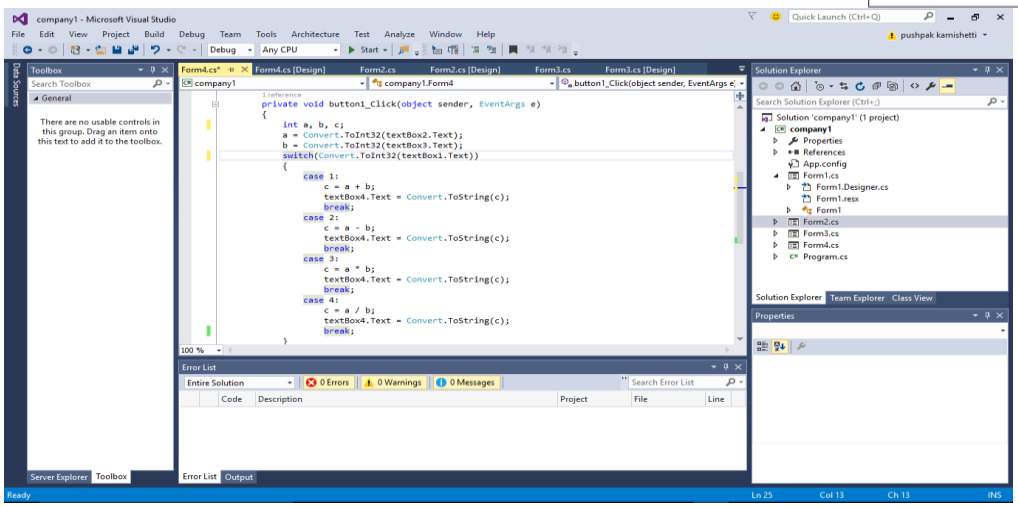

Figure 8output code

The code after clearing the error in the code by using the naming conventions, it got cleared by using punctuation

4.2

In a written report, Analyze actual test results against expected results to identify discrepancies

- Write a sample code in C#
- Give 3 sets of input (For three test cases)
- Identify discrepancies between actual result and test result

Sample code in c#-:

The code I have chosen in this question and the sample code I have done is checking the prime number and code related to it.

Code -:

```
using System;                          //using name space system
namespace pushpak                      //using alternate namespace as pushpak
{
  class one                            //creating class as one
  {
    public static void Main()          //methods starts form there
    {
      Console.Write("Enter a Number : ");          //indicating this sentence to appear on console screen
      int number;                      //class member
      number = Convert.ToInt32(Console.ReadLine());       //converting it to number
      int x;
      x = 0;
      for (int i = 1; i <= number; i++)
      {
        if (number % i == 0)
        {
          x++;
```

```
            }
        }
        if (x == 2)
        {
            Console.WriteLine("Number is a Prime Number and the Largest Factor is
{0}",number);
        }
        else
        {
            Console.WriteLine("Not a Prime Number");
        }
        Console.ReadLine();
    }
  }
}
```

3 sets of different outputs-:

Figure 9first output

Figure 10second output

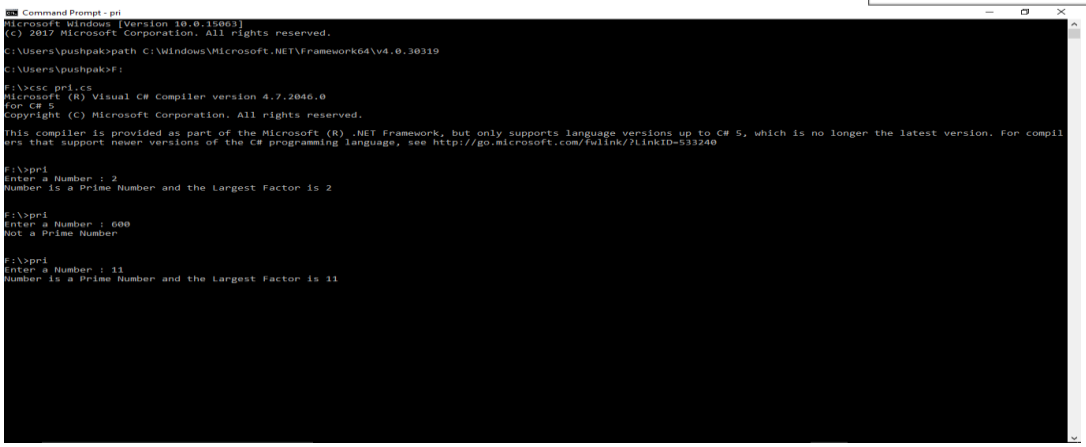

Figure 11third output

Name	User input	Expected output	Current output	Result
Prime number	the user enter number as 2	Should say it is a prime number	As expected it says it is a prime number	Pass
Prime number	The user enter number as 600	Should say it is not a prime number	As expected it says it is not a prime number	Pass
Prime number	The user enter number as 11	Should say it is a prime number	As expected it says it is a prime number	Pass