

Rajshahi Govt. Mohila Polytechnic Institute

Name: shila aktar

Roll: 420642

Technology: CMT

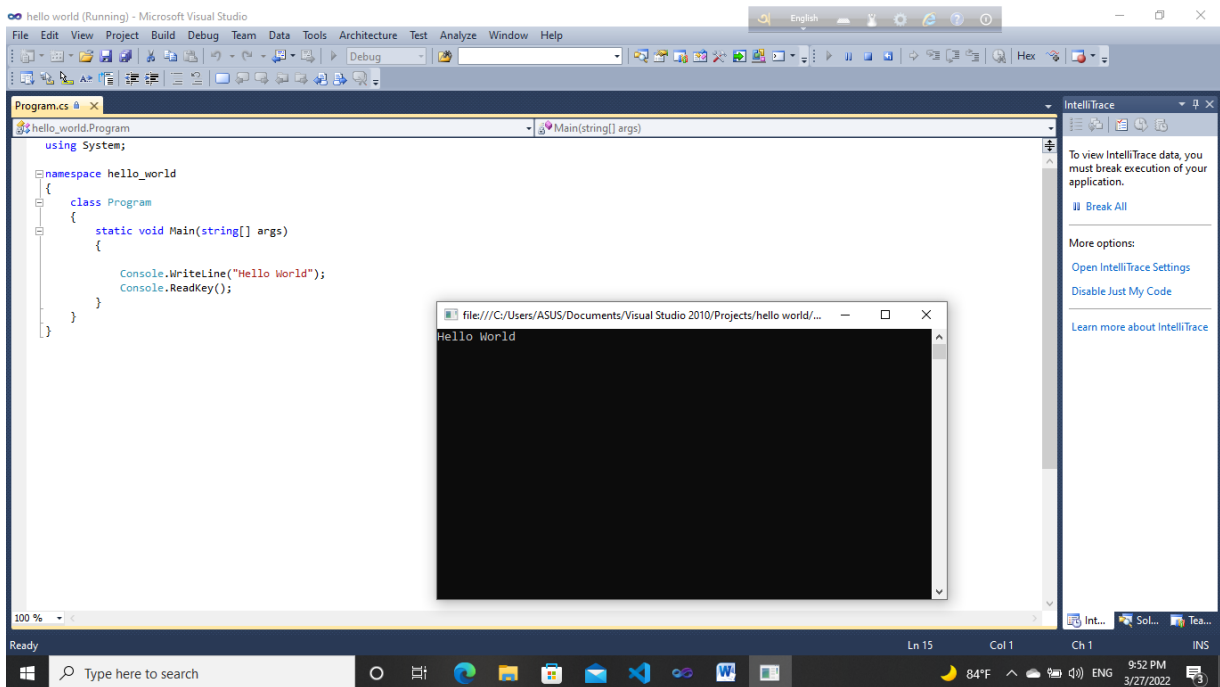
Semester: 4th

Shift: 2nd

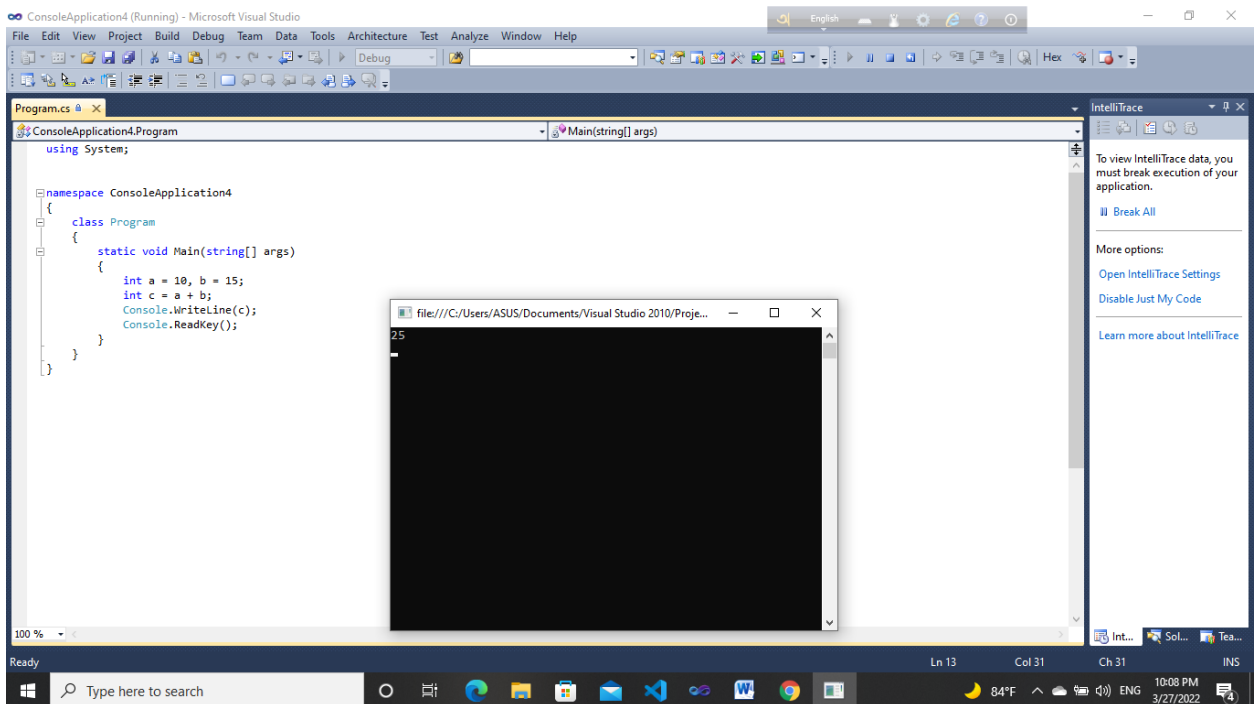
Subject: Object Oriented Programming

Assignment

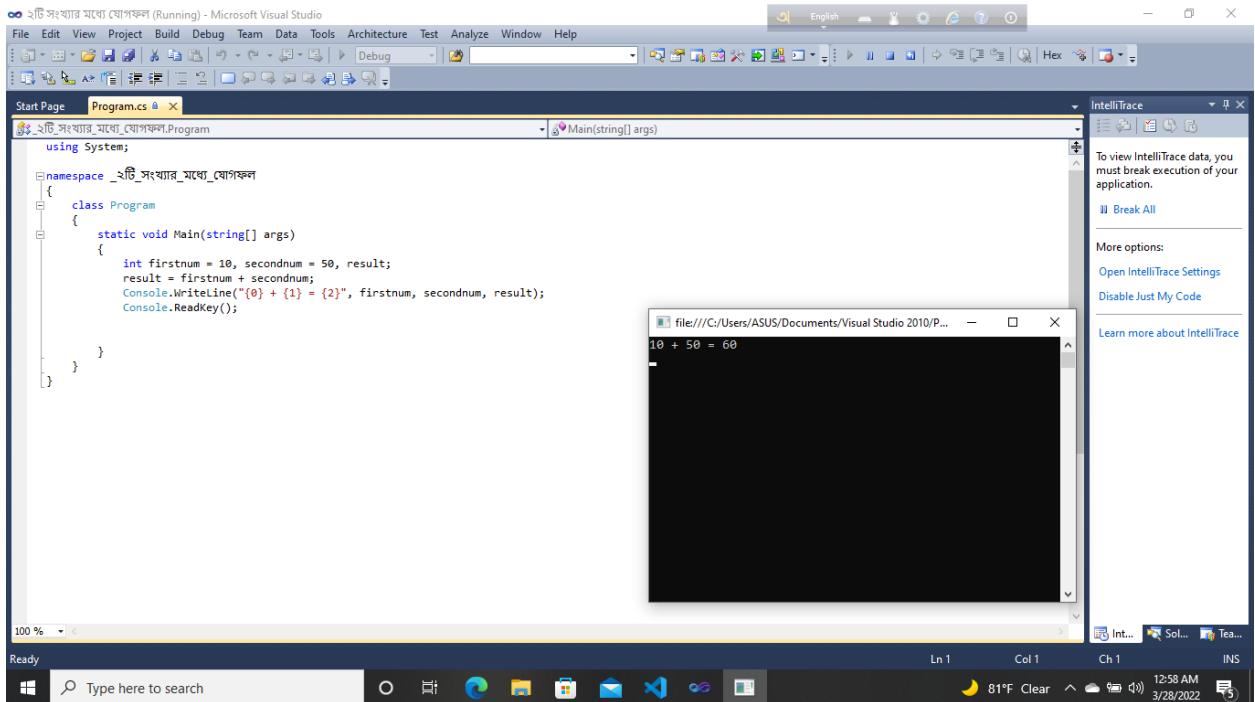
1. Hello world programming



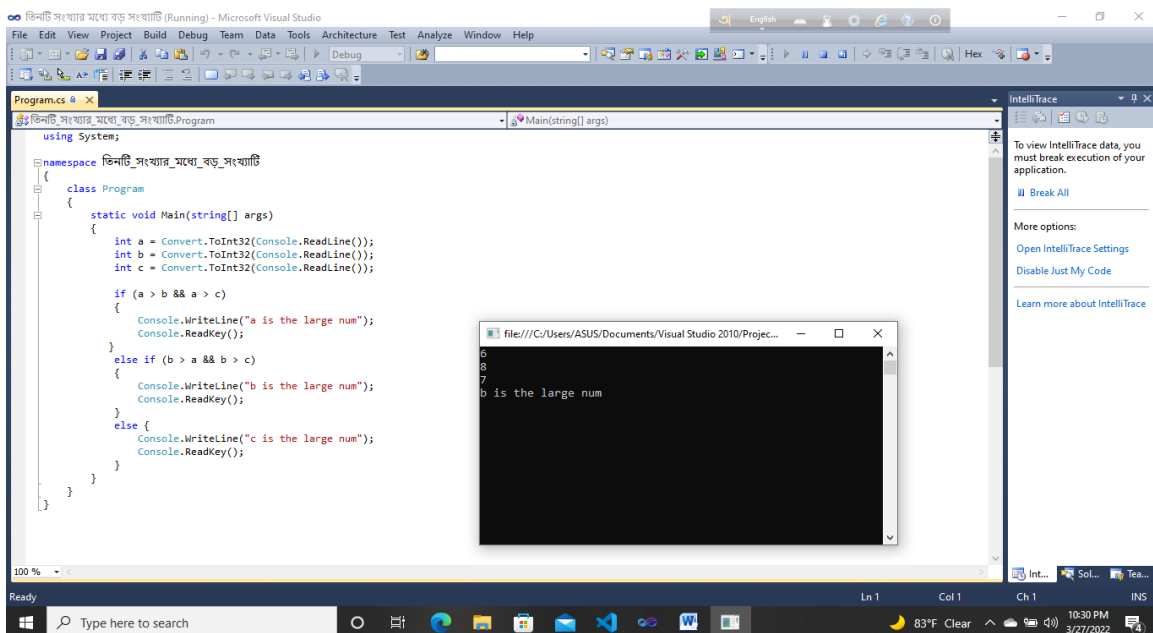
2. দুটি সংখ্যার যোগফলের প্রোগ্রাম..



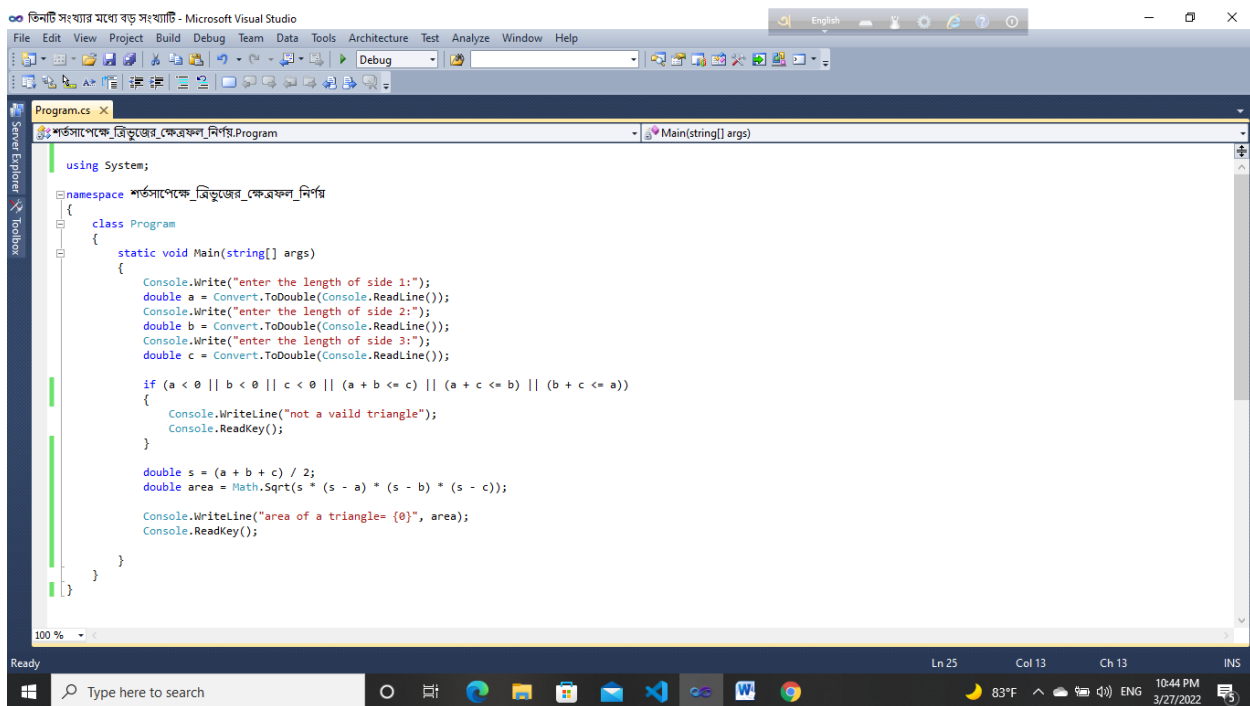
3. ইনপুট নিয়ে দুটি সংখ্যার যোগফল নির্ণয় program...



4. ইনপুট নিয়ে তিনটি সংখ্যার মধ্যে বড় সংখ্যাটি নির্ণয়ের প্রোগ্রাম...



5. শর্তসাপেক্ষ ত্রিভুজের ক্ষেত্রফল নির্য program..



The screenshot shows the Microsoft Visual Studio IDE with a C# program open. The program is titled "শর্তসাপেক্ষ_ত্রিভুজের_ক্ষেত্রফল_নির্য" and is located in the "Program.cs" file. The code defines a namespace, a class, and a static method Main that takes an array of strings as input. It prompts the user to enter the lengths of the three sides of a triangle, converts them to doubles, and then checks if they form a valid triangle using the triangle inequality theorem. If valid, it calculates the area using Heron's formula and prints the result. If not valid, it prints a message indicating that the sides do not form a triangle.

```
using System;

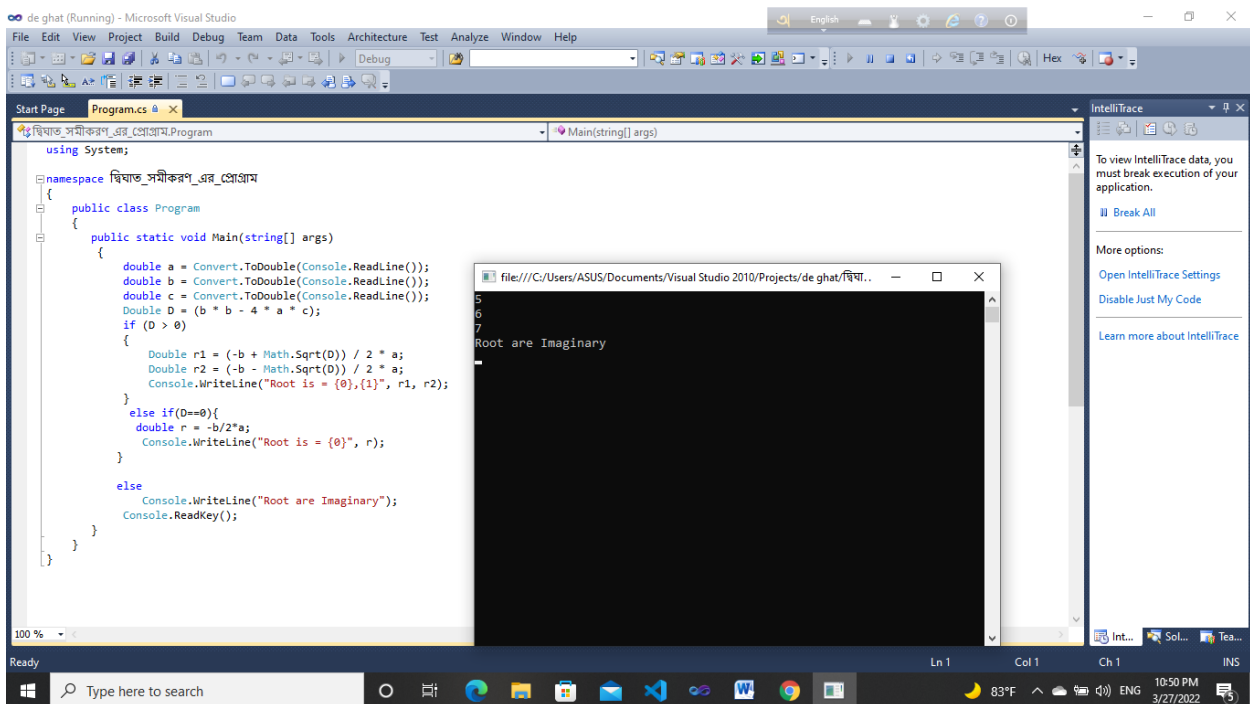
namespace শর্তসাপেক্ষ_ত্রিভুজের_ক্ষেত্রফল_নির্য
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("enter the length of side 1:");
            double a = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine("enter the length of side 2:");
            double b = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine("enter the length of side 3:");
            double c = Convert.ToDouble(Console.ReadLine());

            if (a < 0 || b < 0 || c < 0 || (a + b <= c) || (a + c <= b) || (b + c <= a))
            {
                Console.WriteLine("not a valid triangle");
                Console.ReadKey();
            }

            double s = (a + b + c) / 2;
            double area = Math.Sqrt(s * (s - a) * (s - b) * (s - c));

            Console.WriteLine("area of a triangle= {0}", area);
            Console.ReadKey();
        }
    }
}
```

6. দ্বিঘাত সমীকরণ program..

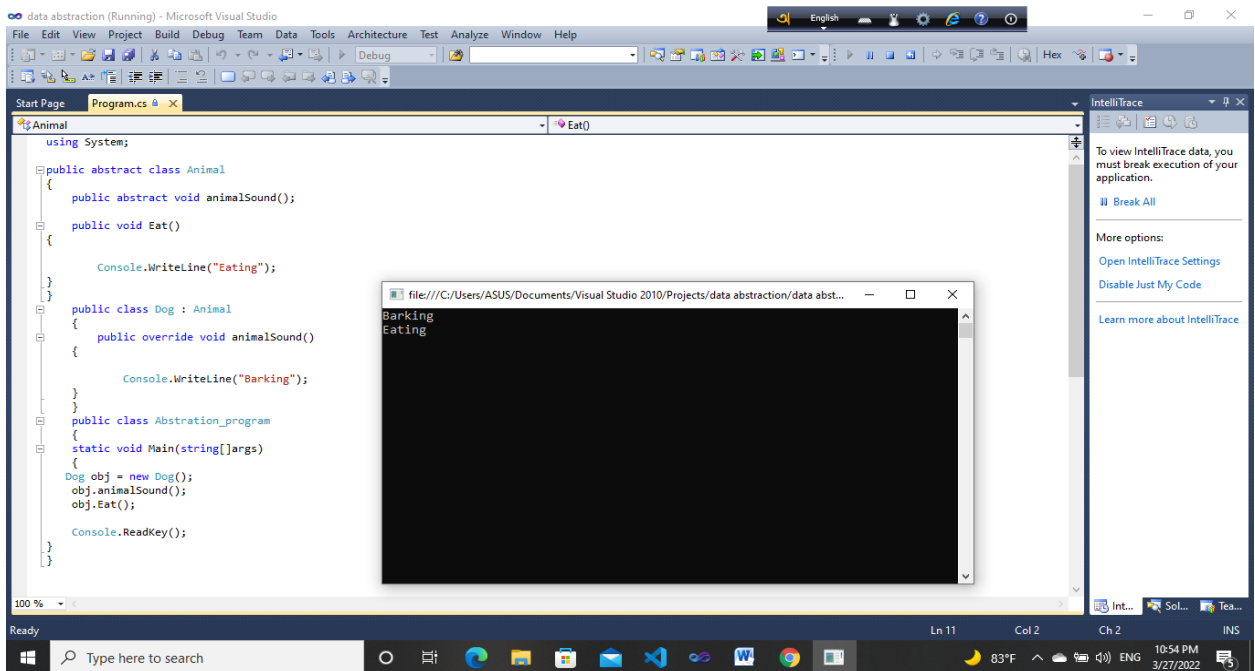


The screenshot shows the Microsoft Visual Studio IDE with a C# program open. The program is titled "দ্বিঘাত সমীকরণ এর প্রোগ্রাম" and is located in the "Program.cs" file. The code defines a namespace, a class, and a static method Main that takes an array of strings as input. It prompts the user to enter the coefficients a, b, and c of a quadratic equation. It then calculates the discriminant D = b^2 - 4ac. If D is greater than 0, it calculates the two real roots r1 and r2 using the quadratic formula and prints them. If D is equal to 0, it calculates the single real root r and prints it. If D is less than 0, it prints a message indicating that the roots are imaginary.

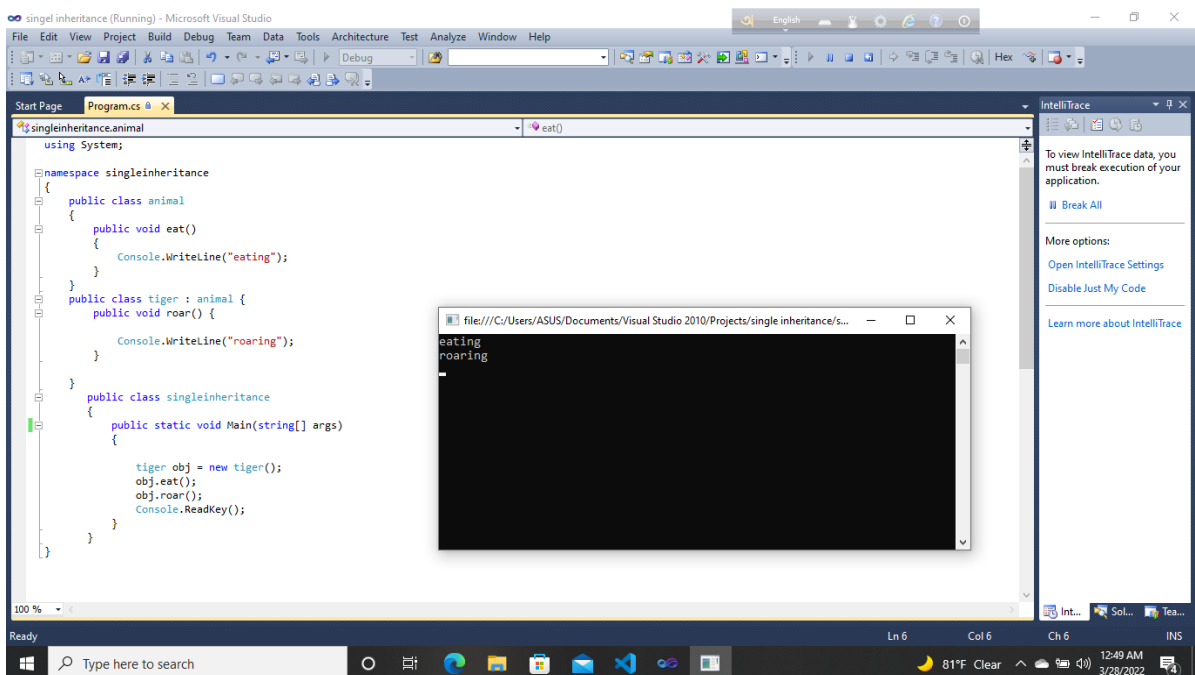
```
using System;

namespace দ্বিঘাত সমীকরণ এর প্রোগ্রাম
{
    public class Program
    {
        public static void Main(string[] args)
        {
            double a = Convert.ToDouble(Console.ReadLine());
            double b = Convert.ToDouble(Console.ReadLine());
            double c = Convert.ToDouble(Console.ReadLine());
            Double D = (b * b - 4 * a * c);
            if (D > 0)
            {
                Double r1 = (-b + Math.Sqrt(D)) / 2 * a;
                Double r2 = (-b - Math.Sqrt(D)) / 2 * a;
                Console.WriteLine("Root is = {0}, {1}", r1, r2);
            }
            else if (D == 0)
            {
                double r = -b / 2 * a;
                Console.WriteLine("Root is = {0}", r);
            }
            else
            {
                Console.WriteLine("Root are Imaginary");
                Console.ReadKey();
            }
        }
    }
}
```

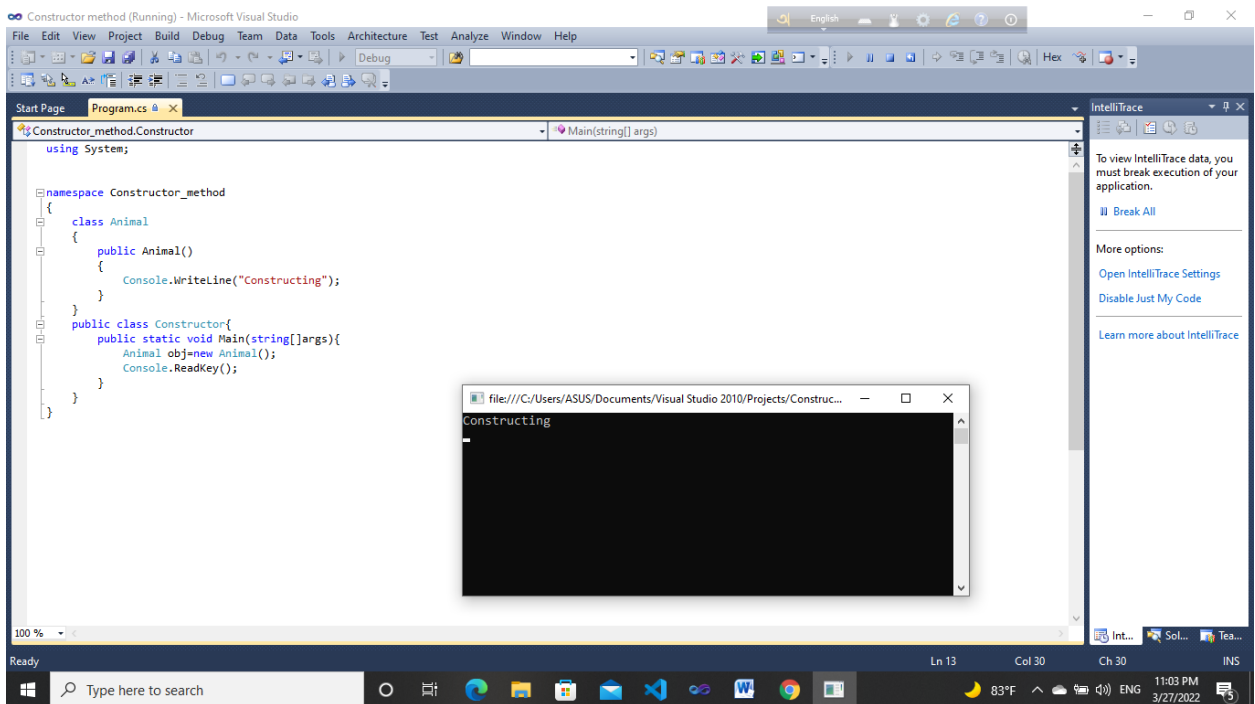
7. Data Abstraction program...



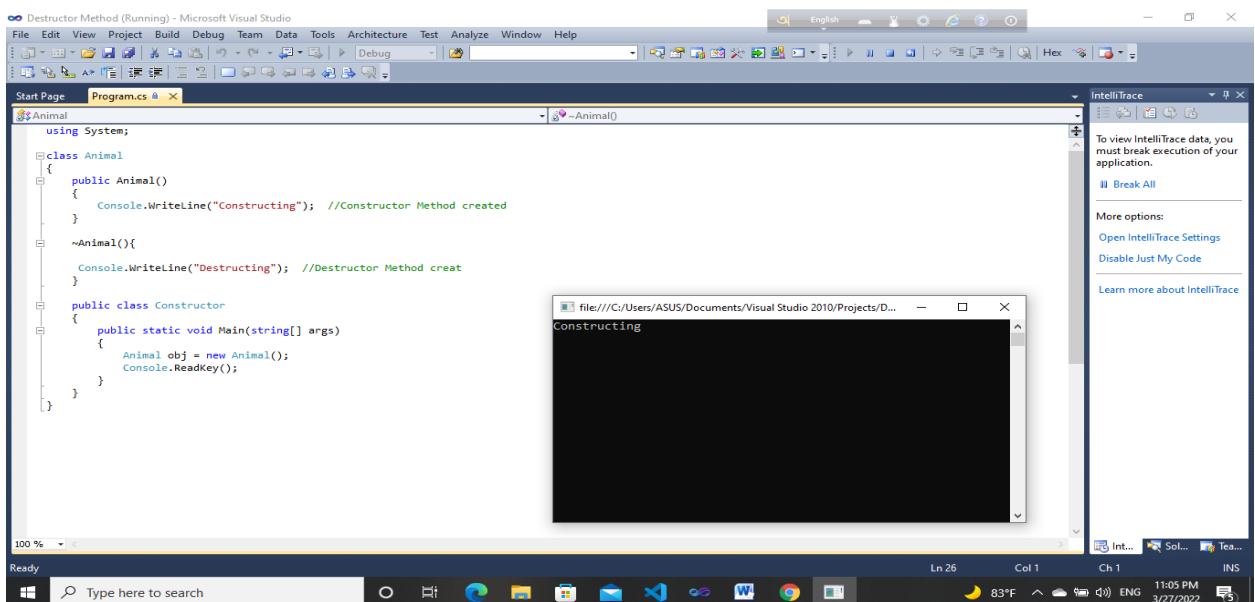
8. Single Inheritance ...



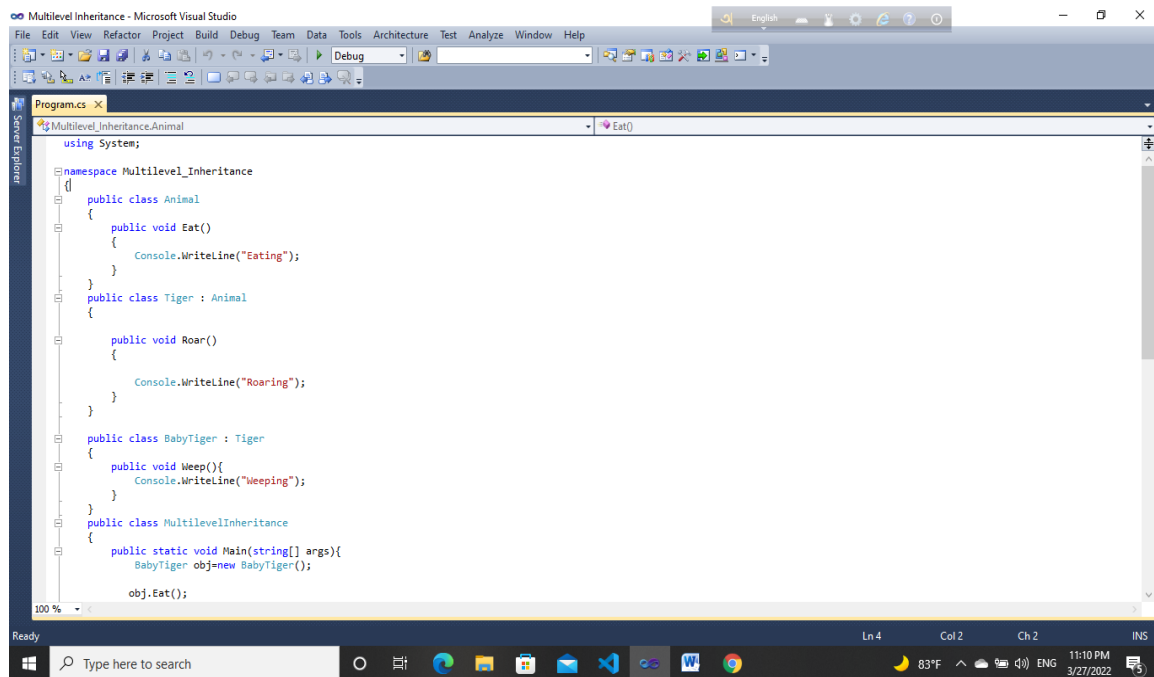
9. Constructor Method....



10. Destructor Method



11. Multilevel Inheritance

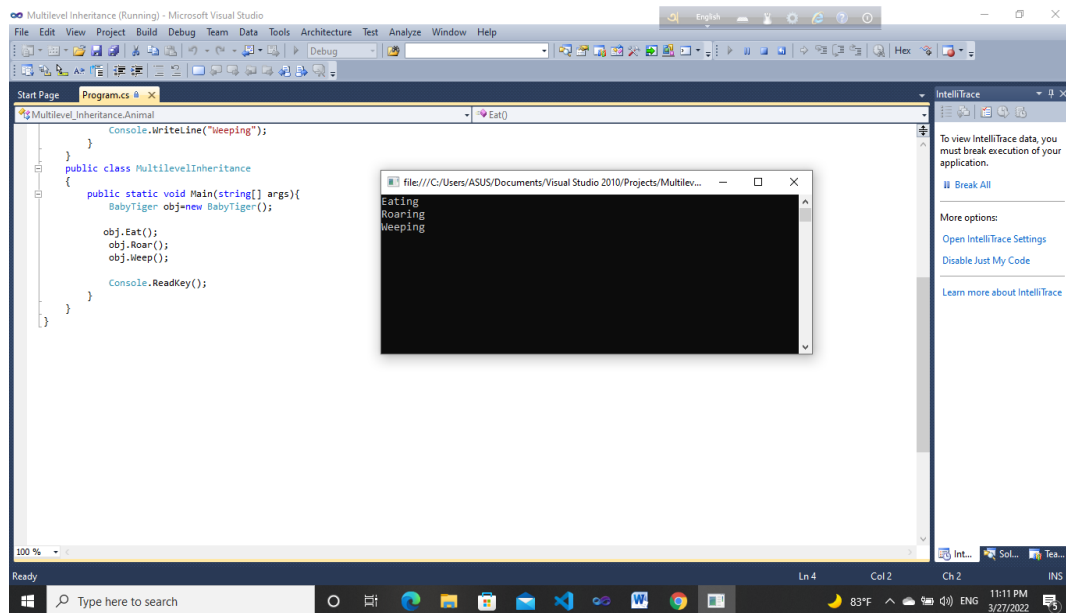


The screenshot shows the Microsoft Visual Studio IDE with a C# project named "Multilevel Inheritance". The code is written in a file named "Program.cs". It defines a namespace "Multilevel_Inheritance" containing three classes: "Animal", "Tiger", and "BabyTiger". "Animal" has an "Eat()" method that prints "Eating". "Tiger" inherits from "Animal" and has a "Roar()" method that prints "Roaring". "BabyTiger" inherits from "Tiger" and has a "Weep()" method that prints "Weeping". A "Main" method creates a "BabyTiger" object and calls its "Eat()", "Roar()", and "Weep()" methods in sequence. The output window shows the results of these calls: "Eating", "Roaring", and "Weeping".

```
using System;

namespace Multilevel_Inheritance
{
    public class Animal
    {
        public void Eat()
        {
            Console.WriteLine("Eating");
        }
    }
    public class Tiger : Animal
    {
        public void Roar()
        {
            Console.WriteLine("Roaring");
        }
    }
    public class BabyTiger : Tiger
    {
        public void Weep()
        {
            Console.WriteLine("Weeping");
        }
    }
    public class MultilevelInheritance
    {
        public static void Main(string[] args)
        {
            BabyTiger obj=new BabyTiger();

            obj.Eat();
        }
    }
}
```



The screenshot shows the Microsoft Visual Studio IDE with the same "Multilevel Inheritance" project. The code is the same as in the previous screenshot. The output window shows the results of the execution: "Eating", "Roaring", and "Weeping". The IntelliTrace window is also visible, showing the execution flow of the program.

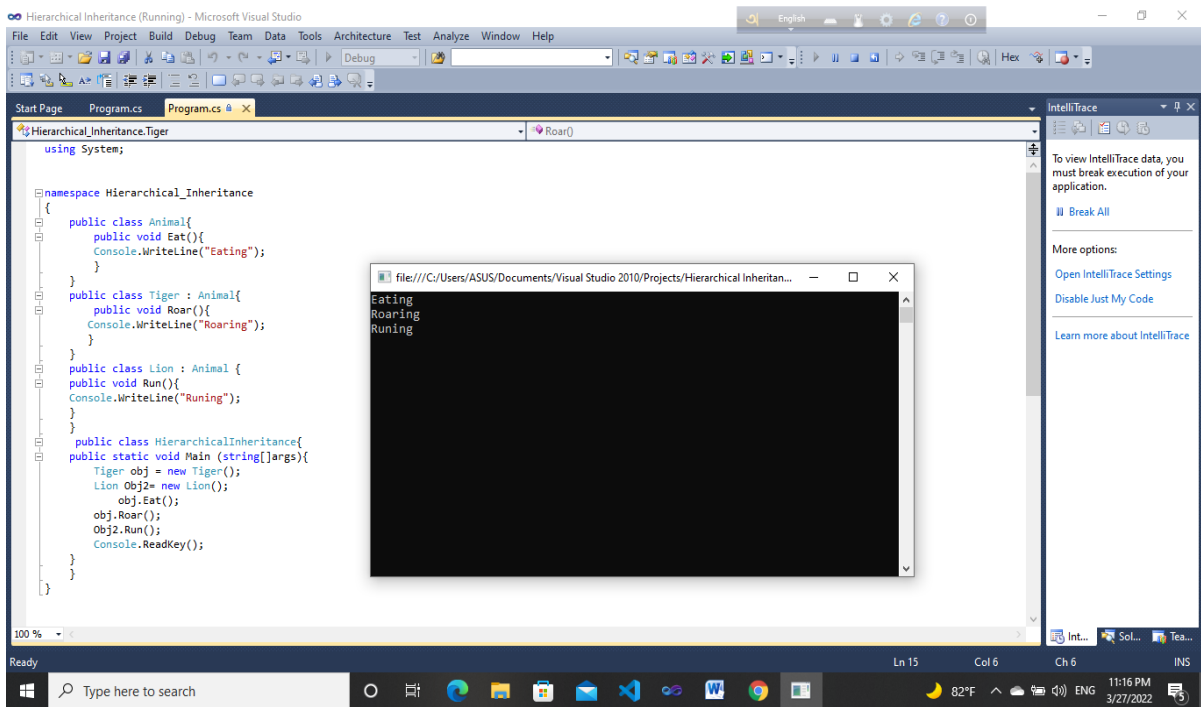
```
using System;

namespace Multilevel_Inheritance
{
    public class Animal
    {
        public void Eat()
        {
            Console.WriteLine("Eating");
        }
    }
    public class Tiger : Animal
    {
        public void Roar()
        {
            Console.WriteLine("Roaring");
        }
    }
    public class BabyTiger : Tiger
    {
        public void Weep()
        {
            Console.WriteLine("Weeping");
        }
    }
    public class MultilevelInheritance
    {
        public static void Main(string[] args)
        {
            BabyTiger obj=new BabyTiger();

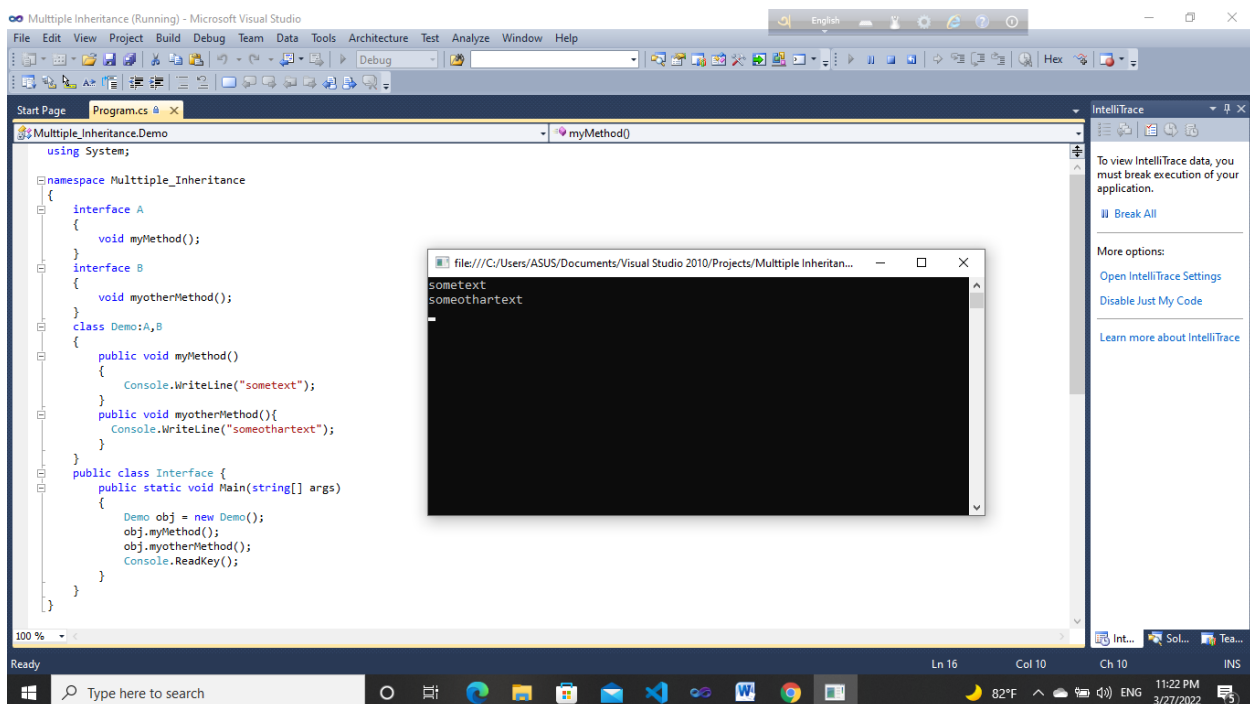
            obj.Eat();
            obj.Roar();
            obj.Weep();

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

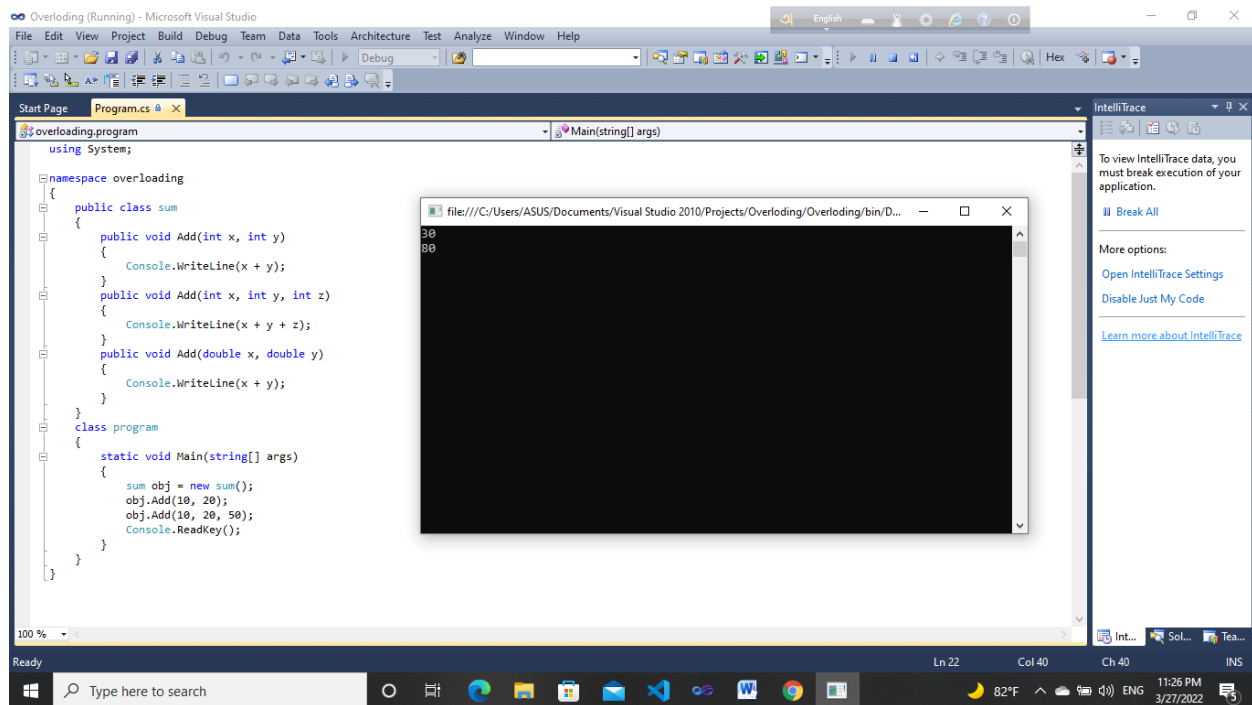
12. Hierarchical Inheritance



13. Multiple Inheritance



14. Overloading Method (Polymorphism)



15. for loop a সিরিজের যোগফল(১+২+৩+.....+১০০)

