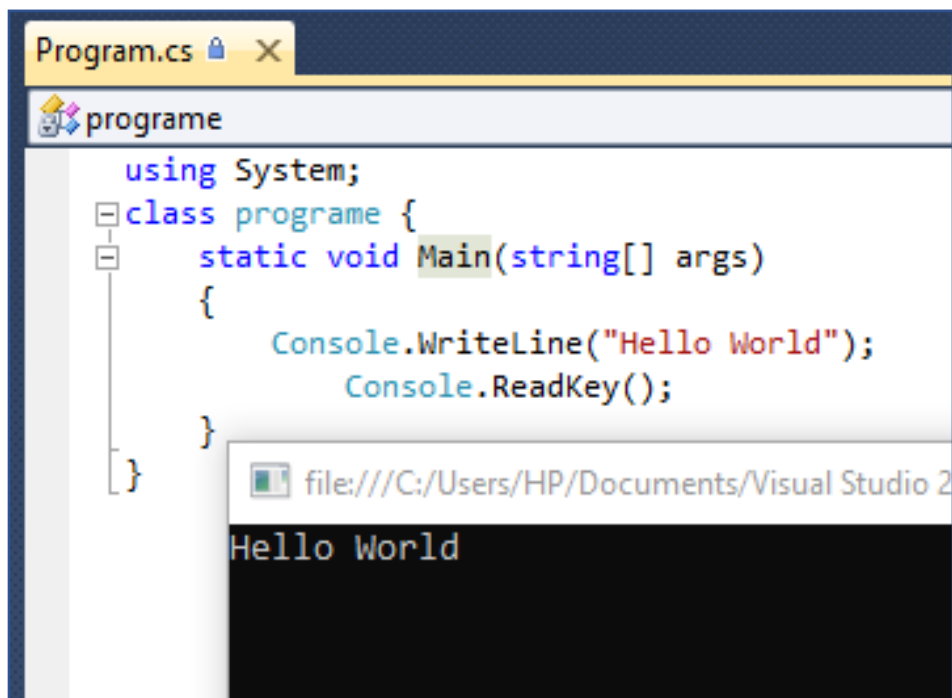


Object Oriented Programming

Airen Akter Orni

CMT 4th, 2nd Shift

1. Hello World Program:

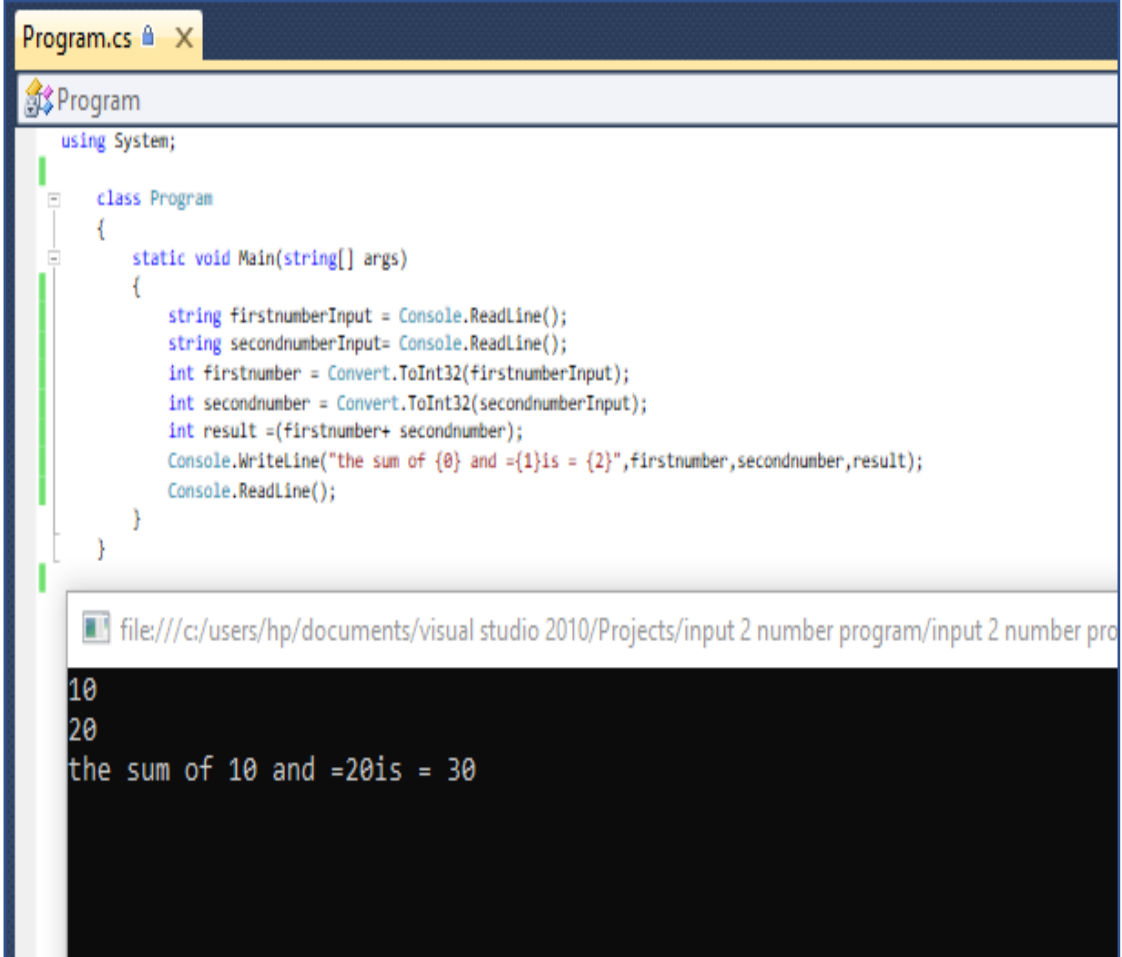


```
Program.cs
programme
using System;
class programme {
    static void Main(string[] args)
    {
        Console.WriteLine("Hello World");
        Console.ReadKey();
    }
}
```

file:///C:/Users/HP/Documents/Visual Studio 2

Hello World

2. Sum between 2 numbers:



The screenshot displays a Visual Studio 2010 IDE window titled "Program.cs". The code is a C# program that reads two numbers from the console, converts them to integers, and calculates their sum. The output window at the bottom shows the program's execution, where the user has entered "10" and "20", and the program has printed "the sum of 10 and =20is = 30".

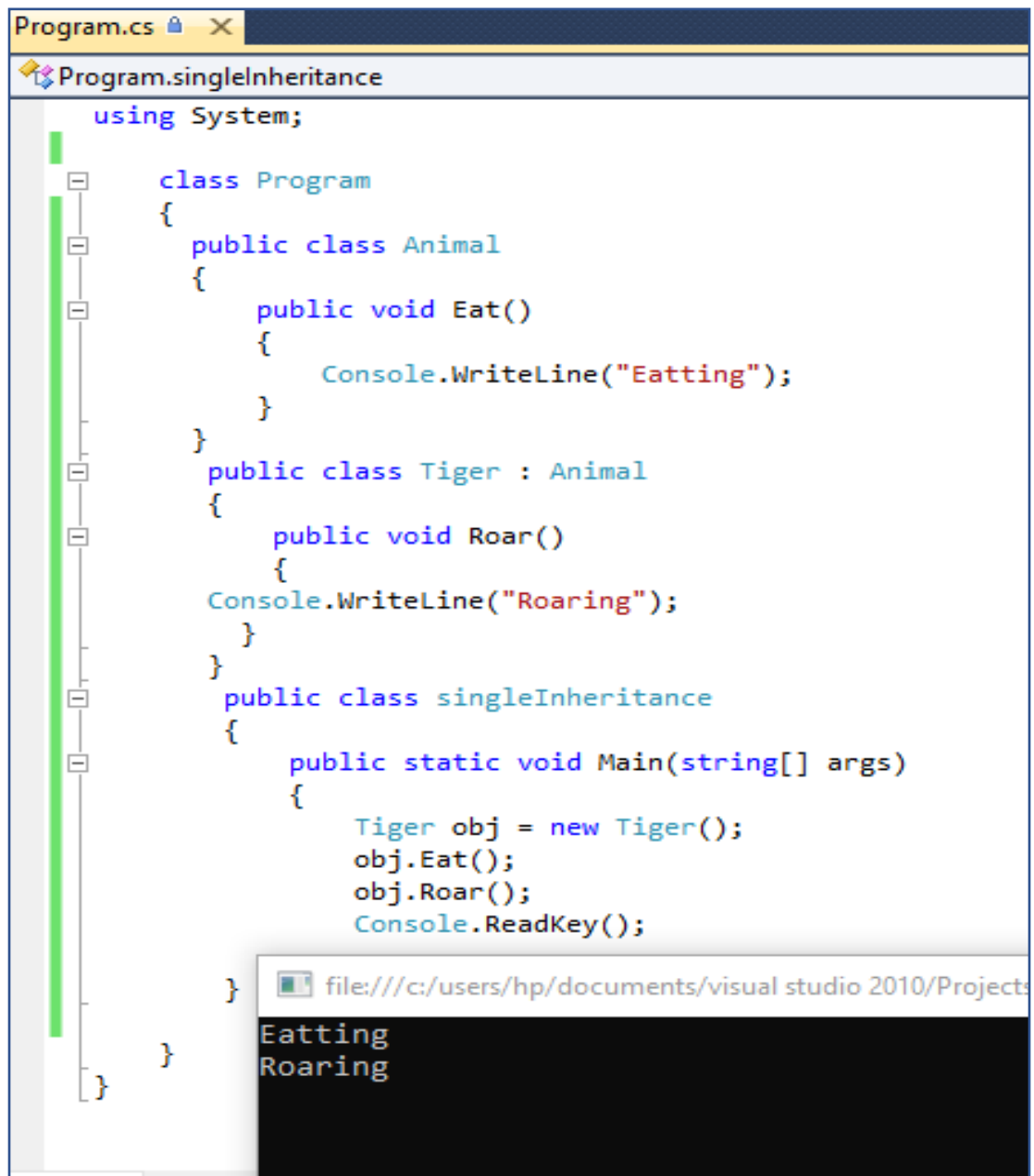
```
using System;

class Program
{
    static void Main(string[] args)
    {
        string firstnumberInput = Console.ReadLine();
        string secondnumberInput = Console.ReadLine();
        int firstnumber = Convert.ToInt32(firstnumberInput);
        int secondnumber = Convert.ToInt32(secondnumberInput);
        int result = (firstnumber + secondnumber);
        Console.WriteLine("the sum of {0} and = {1} is = {2}", firstnumber, secondnumber, result);
        Console.ReadLine();
    }
}
```

file:///c:/users/hp/documents/visual studio 2010/Projects/input 2 number program/input 2 number pro

10
20
the sum of 10 and =20is = 30

3. Single Inheritance:



```
Program.cs
Program.singleInheritance

using System;

class Program
{
    public class Animal
    {
        public void Eat()
        {
            Console.WriteLine("Eating");
        }
    }
    public class Tiger : Animal
    {
        public void Roar()
        {
            Console.WriteLine("Roaring");
        }
    }
    public class singleInheritance
    {
        public static void Main(string[] args)
        {
            Tiger obj = new Tiger();
            obj.Eat();
            obj.Roar();
            Console.ReadKey();
        }
    }
}
```

file:///c:/users/hp/documents/visual studio 2010/Projects/Program.singleInheritance/Program.singleInheritance.csproj

Eating
Roaring

4. Multilevel Inheritance:



```
Program.cs
Program.BabyTiger

using System;

class Program
{
    public class Animal
    {
        public void Eat()
        {
            Console.WriteLine("Eatting.....");
        }
    }
    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();
        }
    }
}
```

file:///c:/users/hp/documents/visual studio 20

Eatting.....
Roaring.....
weeping.....

5. Hairarcical Inheritance:



```
Program.cs
Program.HairarchicalInheritance

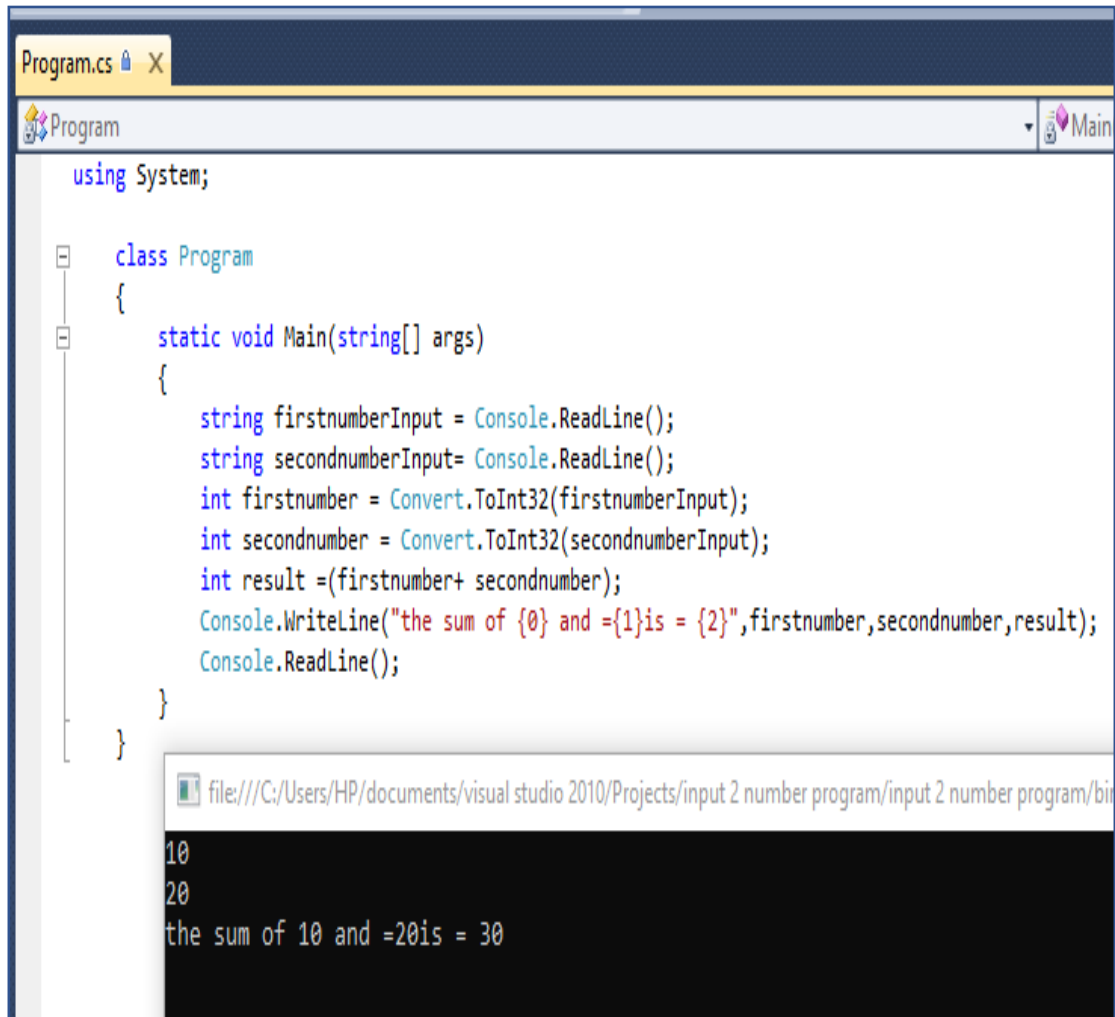
using System;

class Program
{
    public class Animal
    {
        public void Eat()
        {
            Console.WriteLine("Eatting.....");
        }
    }
    public class Tiger : Animal
    {
        public void Roar()
        {
            Console.WriteLine("Roaring.....");
        }
    }
    public class Lion : Animal
    {
        public void Run()
        {
            Console.WriteLine("Running.....");
        }
    }
    public class HairarchicalInheritance
    {
        public static void Main(string[] args)
        {
            Tiger obj = new Tiger();
            Lion obj2 = new Lion();
            obj.Eat();
            obj.Roar();
            obj2.Eat();
            obj2.Run();
            Console.ReadKey();
        }
    }
}
```

file:///c:/users/hp/documents/visual studio 201

Eatting.....
Roaring.....
Eatting.....
Running.....

6. Sum between 2 numbers:



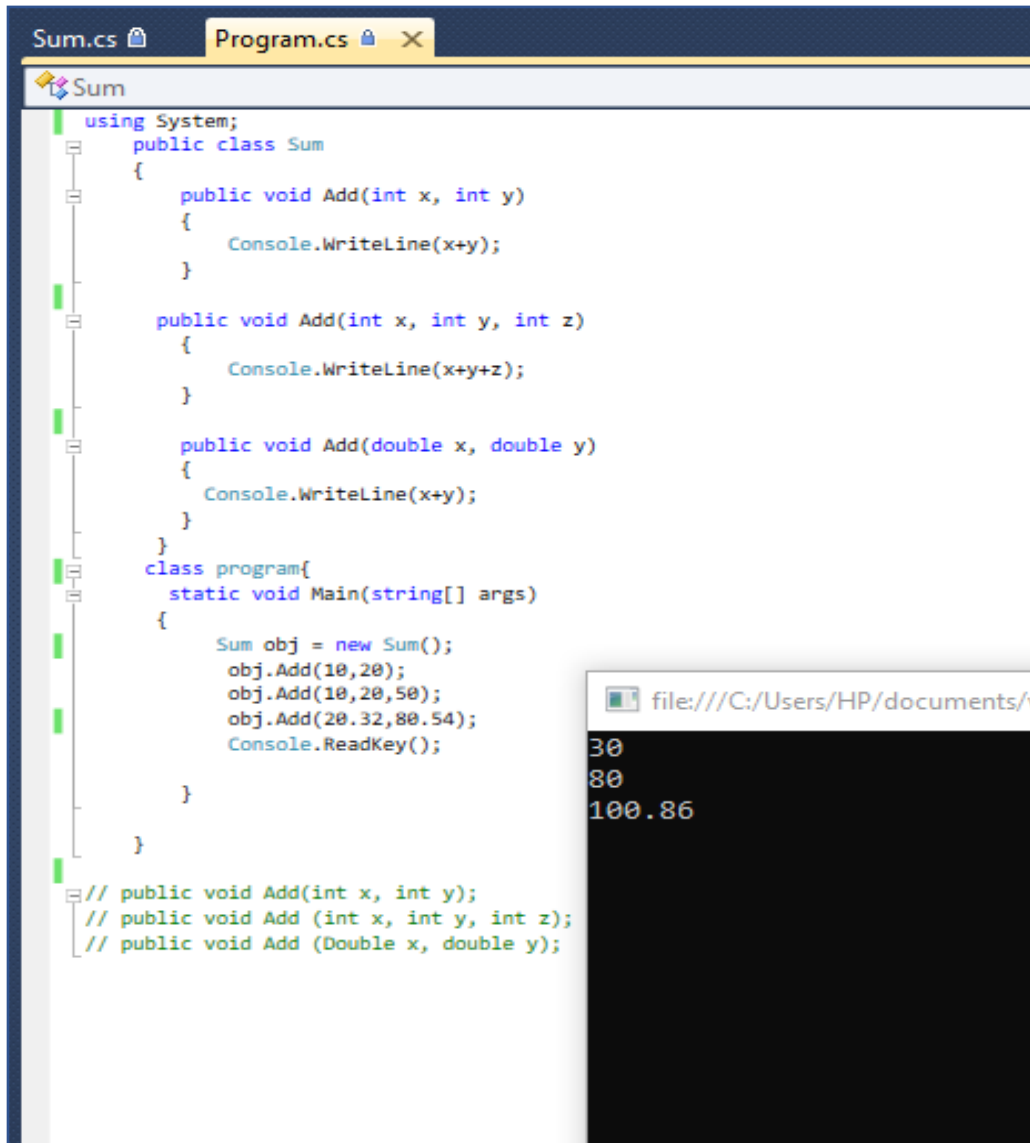
```
Program.cs X
Program
using System;

class Program
{
    static void Main(string[] args)
    {
        string firstnumberInput = Console.ReadLine();
        string secondnumberInput= Console.ReadLine();
        int firstnumber = Convert.ToInt32(firstnumberInput);
        int secondnumber = Convert.ToInt32(secondnumberInput);
        int result =(firstnumber+ secondnumber);
        Console.WriteLine("the sum of {0} and = {1}is = {2}",firstnumber,secondnumber,result);
        Console.ReadLine();
    }
}
```

file:///C:/Users/HP/documents/visual studio 2010/Projects/input 2 number program/input 2 number program/bin

10
20
the sum of 10 and =20is = 30

7. Polymorphisum:



```
Sum.cs Program.cs X
Sum
using System;
public class Sum
{
    public void Add(int x, int y)
    {
        Console.WriteLine(x+y);
    }

    public void Add(int x, int y, int z)
    {
        Console.WriteLine(x+y+z);
    }

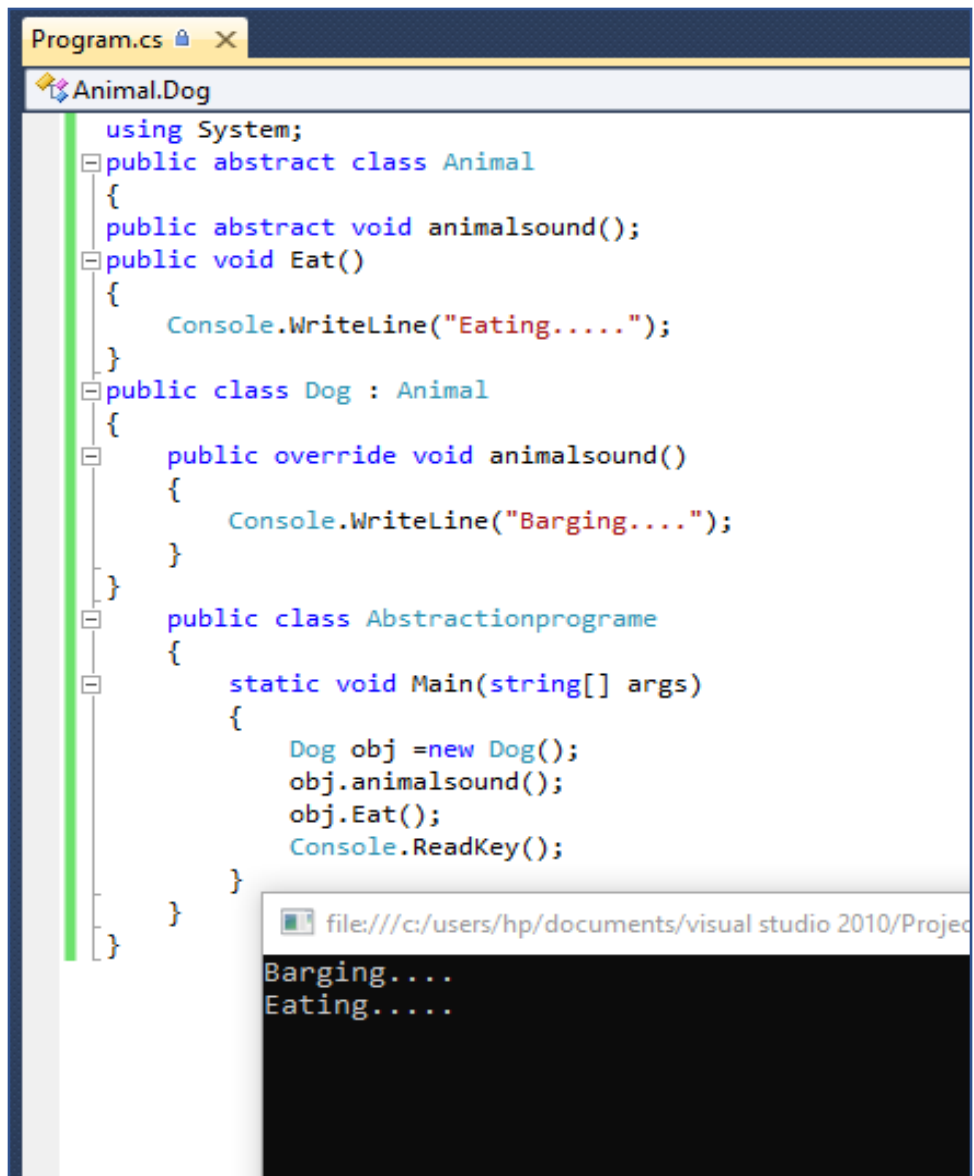
    public void Add(double x, double y)
    {
        Console.WriteLine(x+y);
    }
}
class program{
    static void Main(string[] args)
    {
        Sum obj = new Sum();
        obj.Add(10,20);
        obj.Add(10,20,50);
        obj.Add(20.32,80.54);
        Console.ReadKey();
    }
}

// public void Add(int x, int y);
// public void Add (int x, int y, int z);
// public void Add (Double x, double y);
```

file:///C:/Users/HP/documents/

30
80
100.86

8. Abstraction programme:



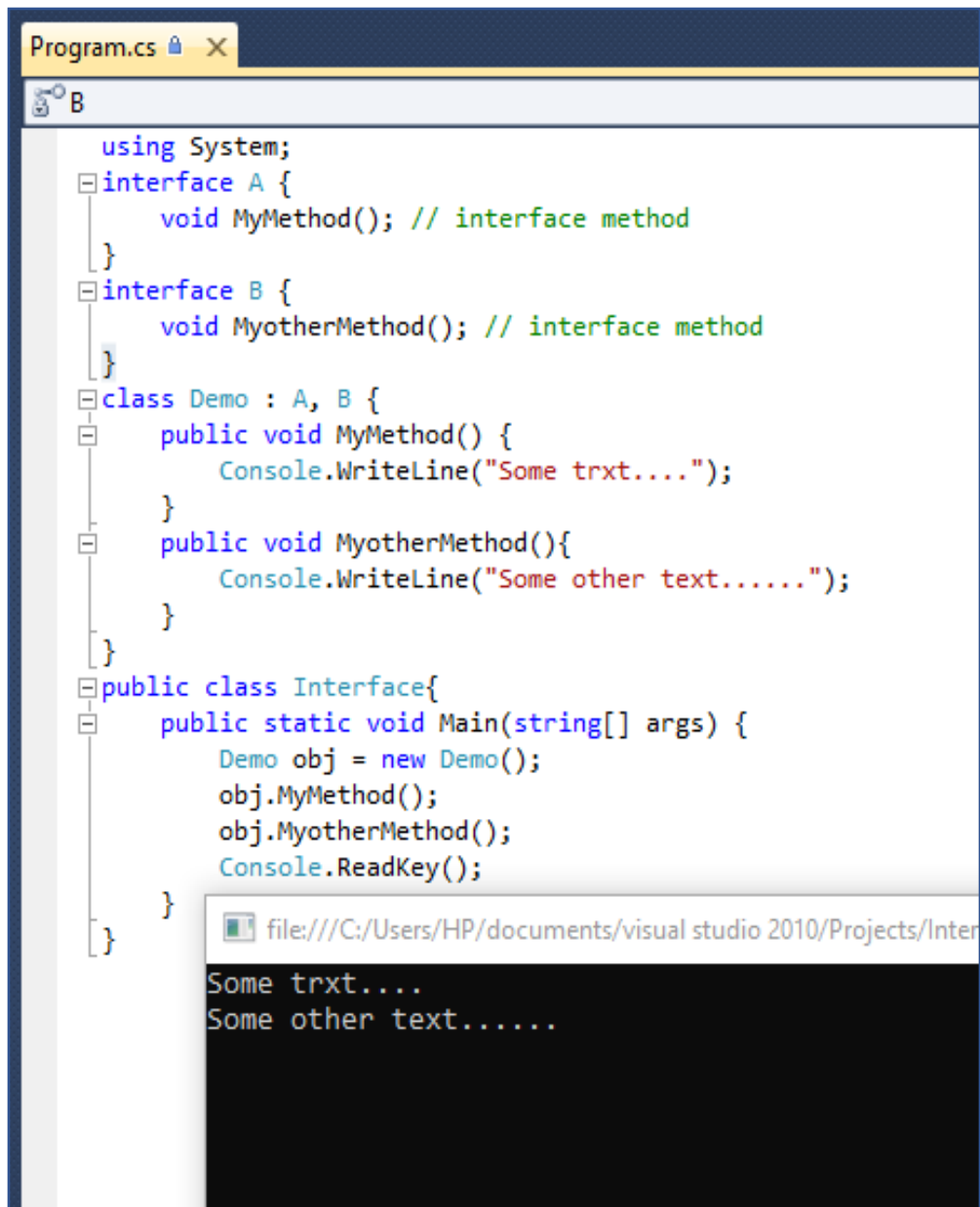
```
Program.cs
Animal.Dog

using System;
public abstract class Animal
{
    public abstract void animalsound();
    public void Eat()
    {
        Console.WriteLine("Eating.....");
    }
}
public class Dog : Animal
{
    public override void animalsound()
    {
        Console.WriteLine("Barging....");
    }
}
public class Abstractionprograme
{
    static void Main(string[] args)
    {
        Dog obj = new Dog();
        obj.animalsound();
        obj.Eat();
        Console.ReadKey();
    }
}
```

file:///c:/users/hp/documents/visual studio 2010/Projec

Barging....
Eating.....

9. Multiple Inheritance:

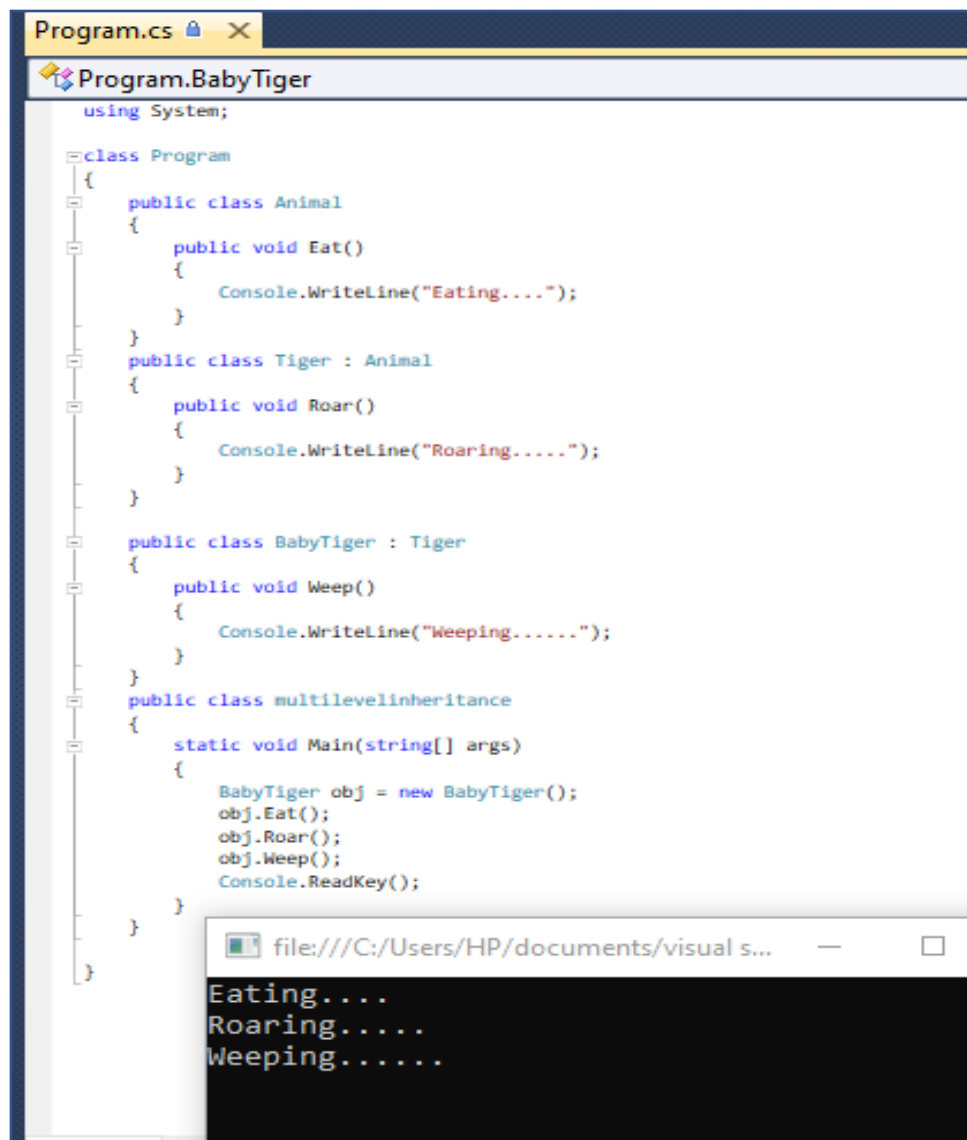


```
Program.cs X
B
using System;
interface A {
    void MyMethod(); // interface method
}
interface B {
    void MyotherMethod(); // interface method
}
class Demo : A, B {
    public void MyMethod() {
        Console.WriteLine("Some trxt....");
    }
    public void MyotherMethod(){
        Console.WriteLine("Some other text.....");
    }
}
public class Interface{
    public static void Main(string[] args) {
        Demo obj = new Demo();
        obj.MyMethod();
        obj.MyotherMethod();
        Console.ReadKey();
    }
}
```

file:///C:/Users/HP/documents/visual studio 2010/Projects/Inter

Some trxt....
Some other text.....

10. Multilevel Inheritance:



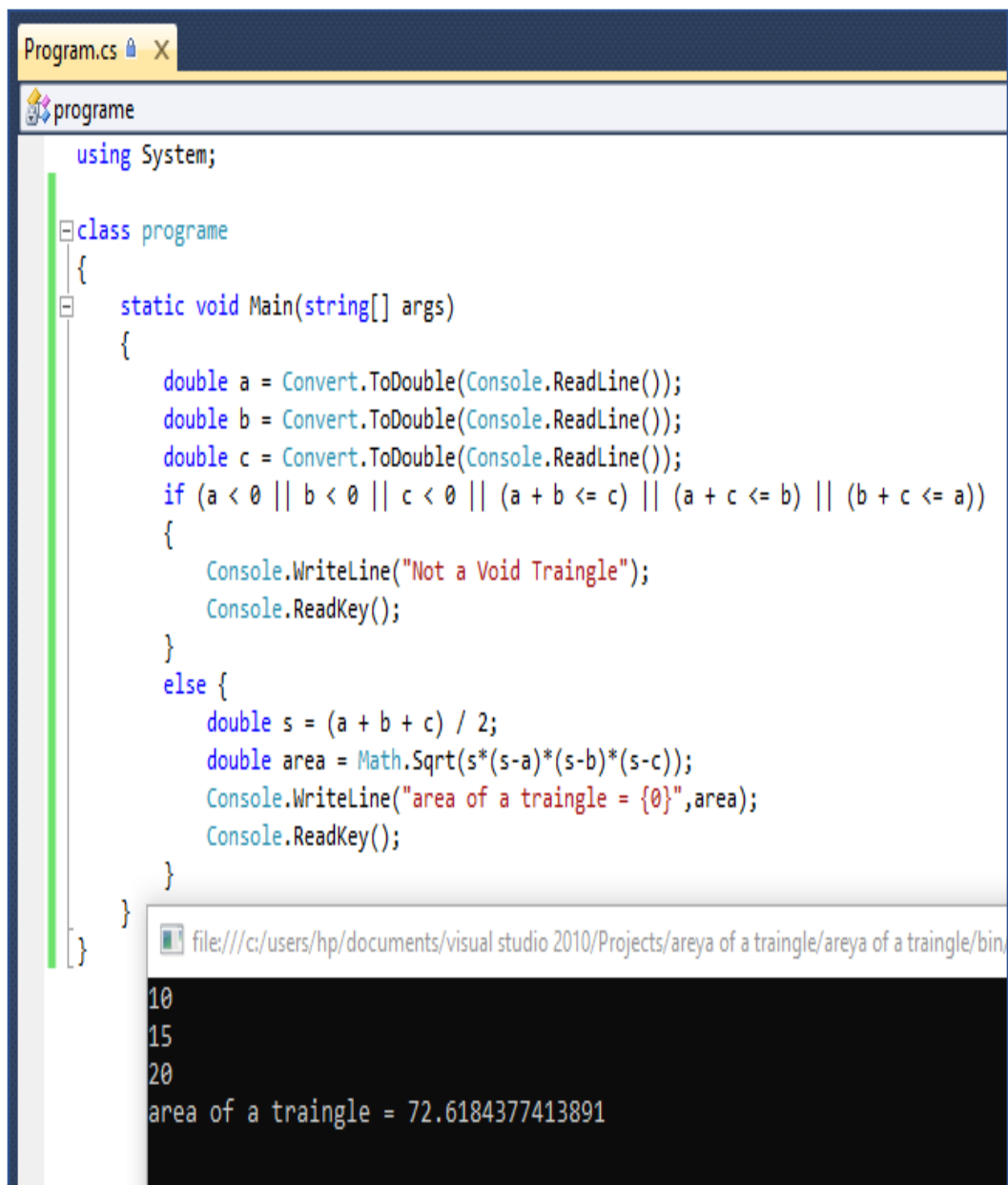
```
Program.cs
Program.BabyTiger

using System;

class Program
{
    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
    {
        static void Main(string[] args)
        {
            BabyTiger obj = new BabyTiger();
            obj.Eat();
            obj.Roar();
            obj.Weep();
            Console.ReadKey();
        }
    }
}
```

file:///C:/Users/HP/documents/visual s...
Eating....
Roaring.....
Weeping.....

11. Area of a triangle:



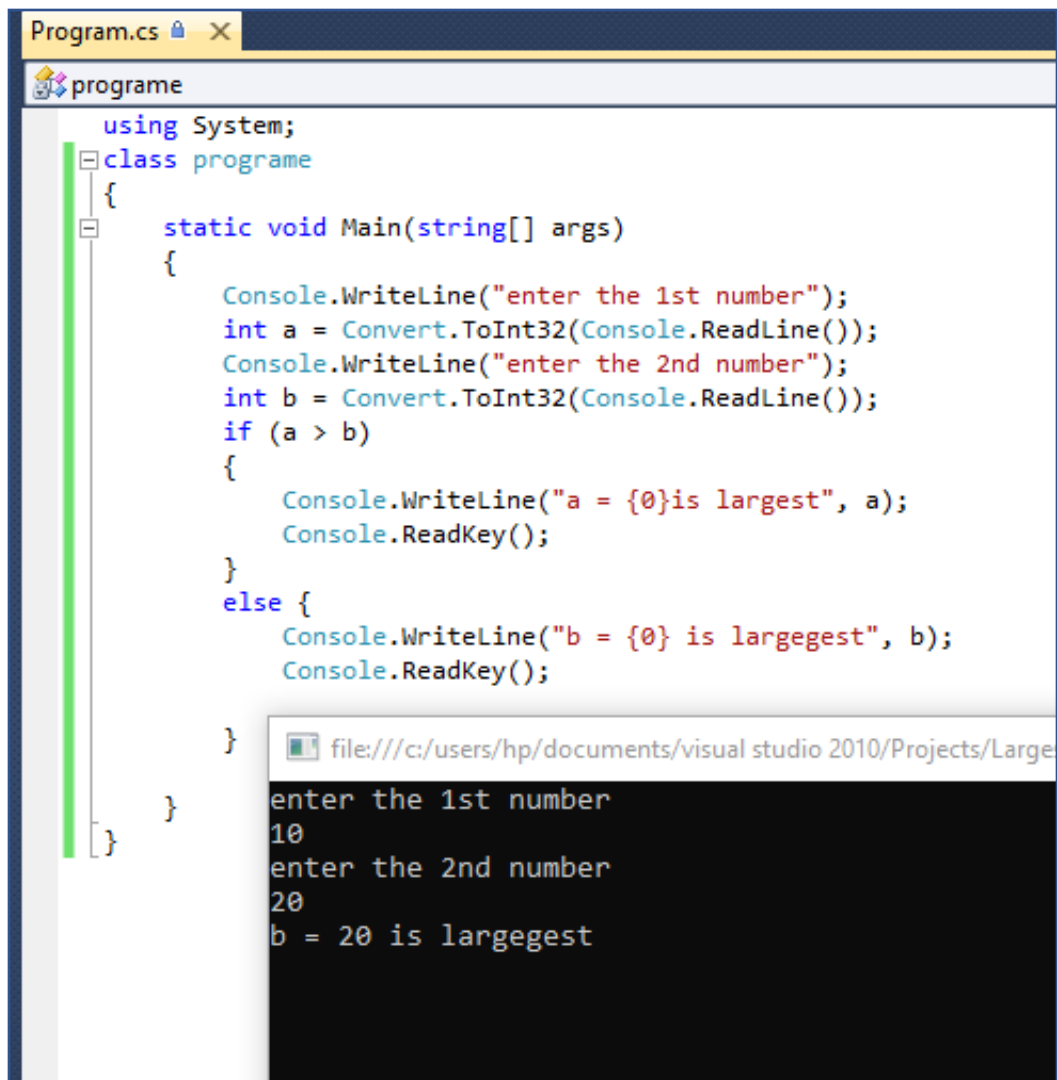
```
Program.cs X
program
using System;

class program
{
    static void Main(string[] args)
    {
        double a = Convert.ToDouble(Console.ReadLine());
        double b = Convert.ToDouble(Console.ReadLine());
        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();
        }
        else {
            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();
        }
    }
}
```

file:///c:/users/hp/documents/visual studio 2010/Projects/area of a triangle/area of a triangle/bin/

10
15
20
area of a triangle = 72.6184377413891

12. Largest number between 2 numbers:



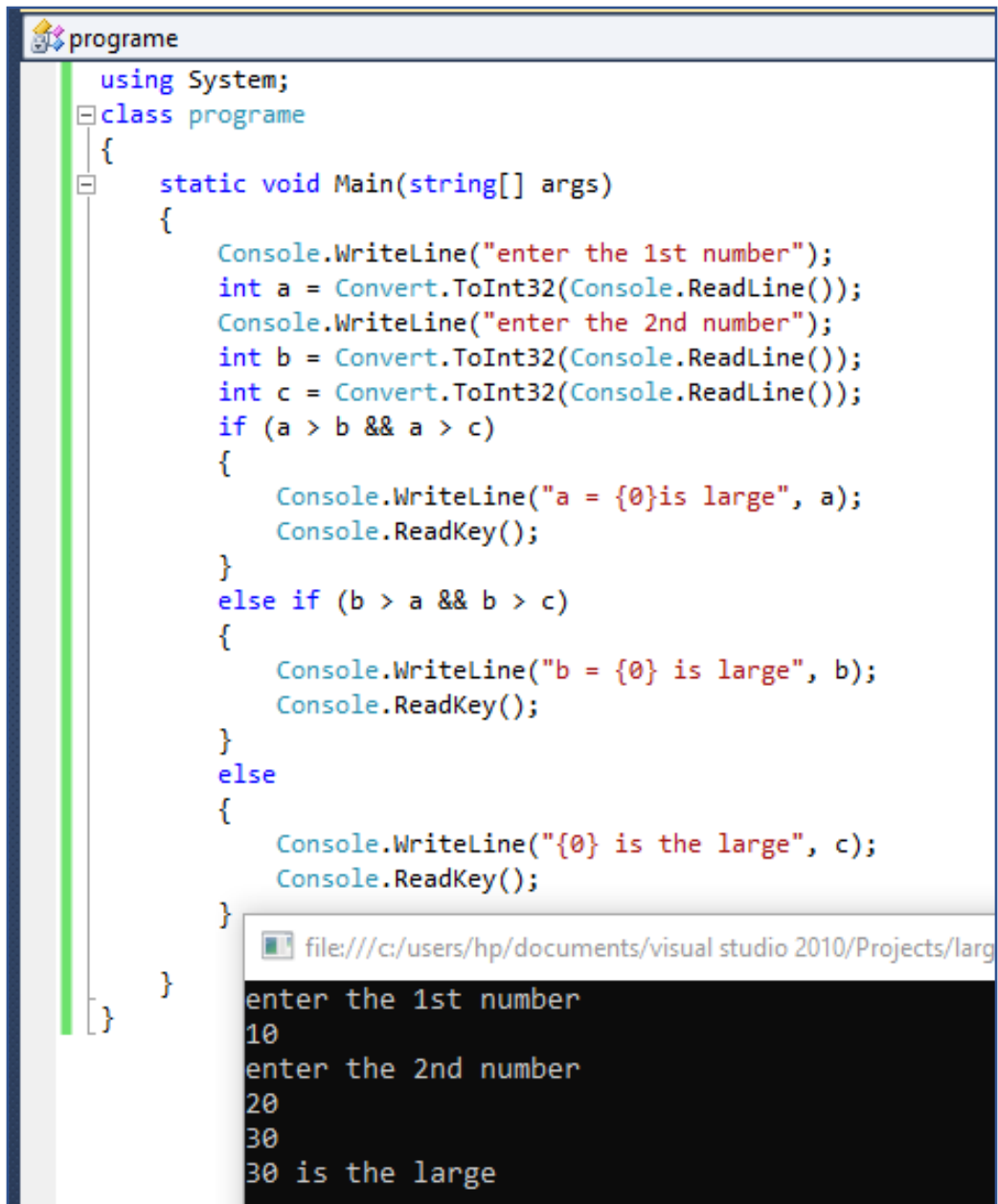
The image shows a screenshot of the Visual Studio 2010 IDE. The main window displays a C# file named 'Program.cs' with the following code:

```
using System;
class programme
{
    static void Main(string[] args)
    {
        Console.WriteLine("enter the 1st number");
        int a = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("enter the 2nd number");
        int b = Convert.ToInt32(Console.ReadLine());
        if (a > b)
        {
            Console.WriteLine("a = {0}is largest", a);
            Console.ReadKey();
        }
        else {
            Console.WriteLine("b = {0} is largegest", b);
            Console.ReadKey();
        }
    }
}
```

Below the code editor, a console window is open, showing the program's execution. The output is as follows:

```
enter the 1st number
10
enter the 2nd number
20
b = 20 is largegest
```

13. Largest number between 3 numbers:

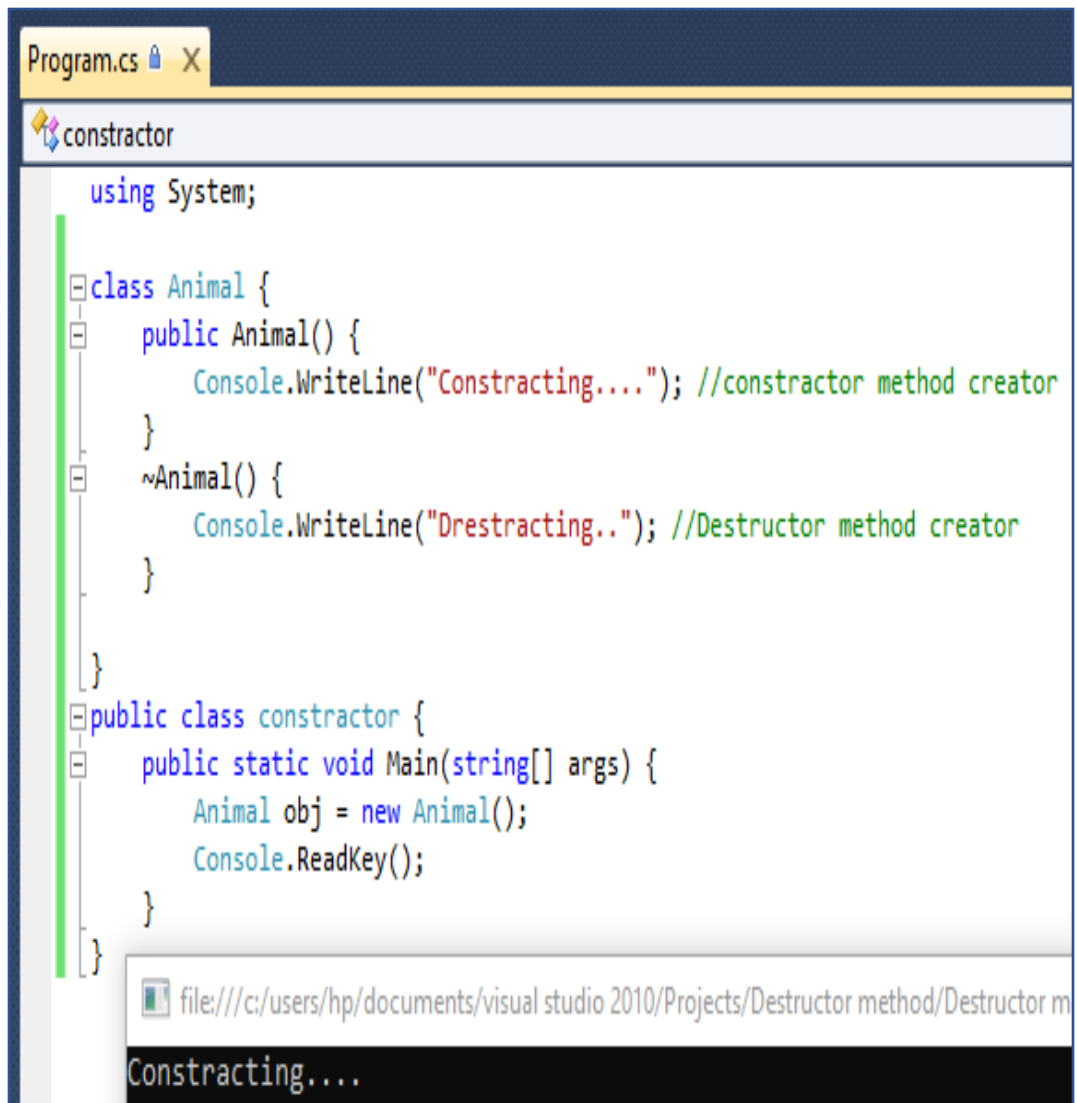


```
using System;
class programe
{
    static void Main(string[] args)
    {
        Console.WriteLine("enter the 1st number");
        int a = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("enter the 2nd number");
        int b = Convert.ToInt32(Console.ReadLine());
        int c = Convert.ToInt32(Console.ReadLine());
        if (a > b && a > c)
        {
            Console.WriteLine("a = {0} is large", a);
            Console.ReadKey();
        }
        else if (b > a && b > c)
        {
            Console.WriteLine("b = {0} is large", b);
            Console.ReadKey();
        }
        else
        {
            Console.WriteLine("{0} is the large", c);
            Console.ReadKey();
        }
    }
}
```

file:///c:/users/hp/documents/visual studio 2010/Projects/larg

```
enter the 1st number
10
enter the 2nd number
20
30
30 is the large
```

14. Constractor -Distractor method:



```
Program.cs X
constractor

using System;

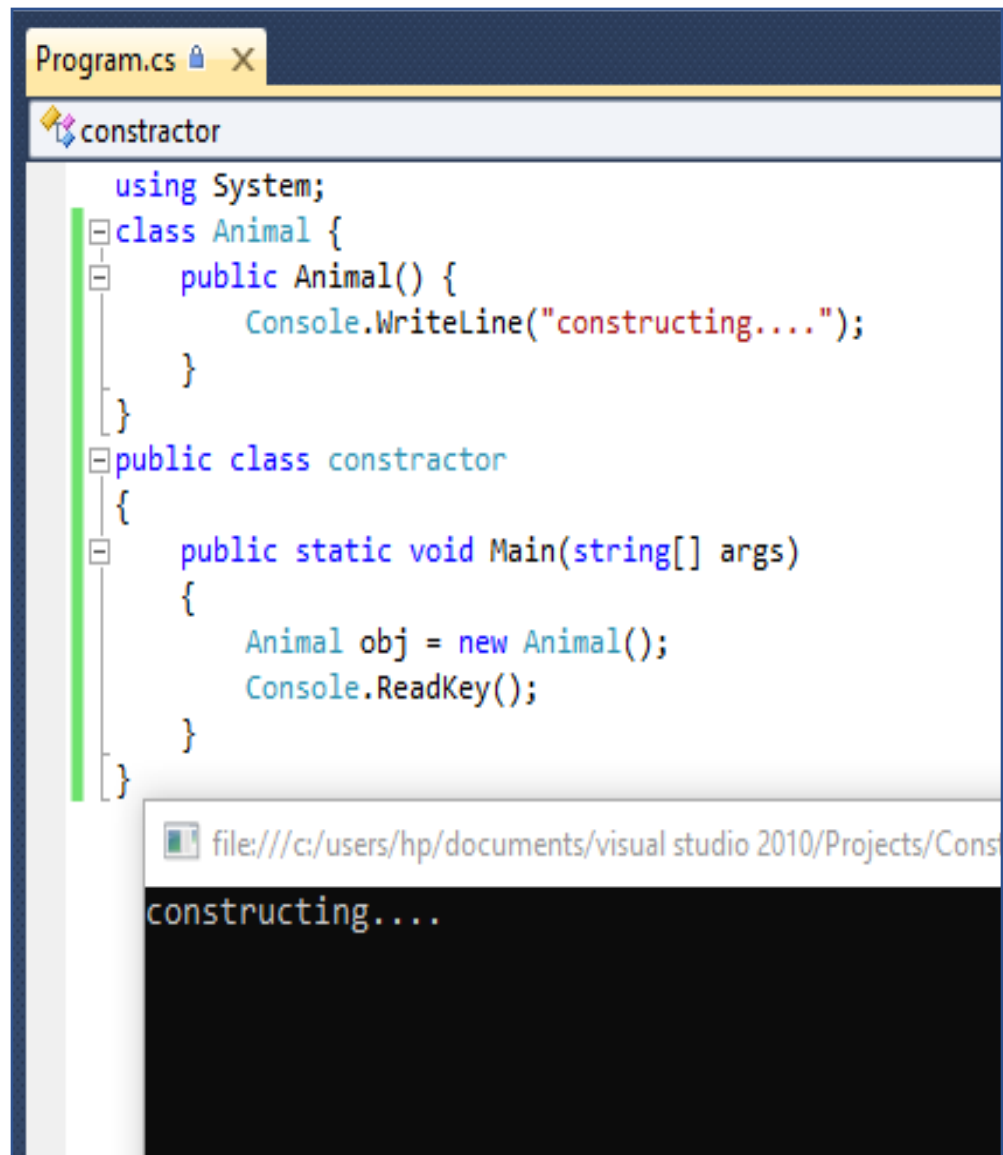
class Animal {
    public Animal() {
        Console.WriteLine("Constracting...."); //constractor method creator
    }
    ~Animal() {
        Console.WriteLine("Drestracting.."); //Destructor method creator
    }
}

public class constractor {
    public static void Main(string[] args) {
        Animal obj = new Animal();
        Console.ReadKey();
    }
}
```

file:///c:/users/hp/documents/visual studio 2010/Projects/Destructor method/Destructor m

Constracting....

15. Constractor method:



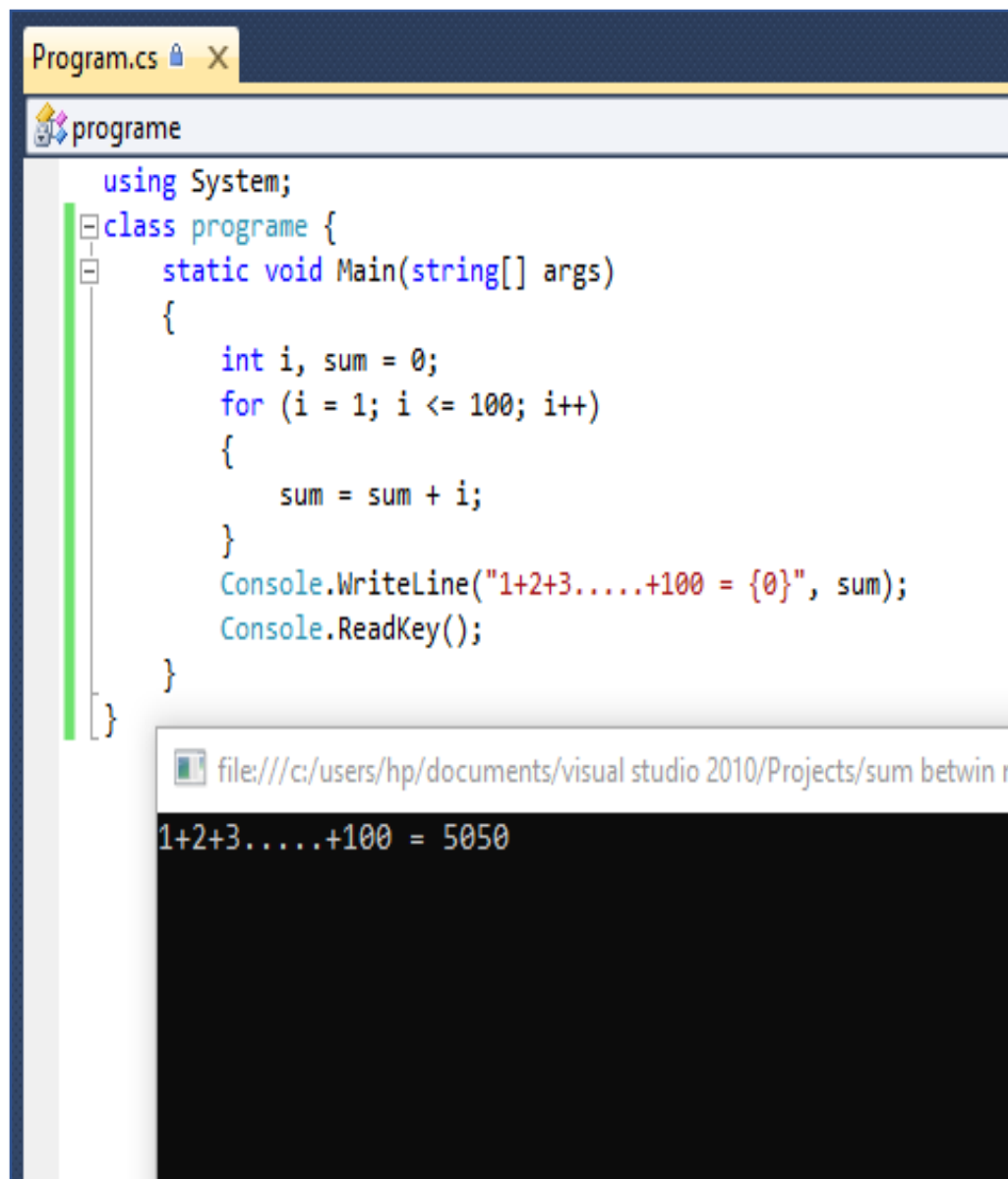
The screenshot shows a Visual Studio 2010 window with a file named 'Program.cs'. The code defines a class 'Animal' with a constructor that prints 'constructing....' and a 'Main' method that creates an 'Animal' object and reads a key. The output window at the bottom shows the text 'constructing....'.

```
Program.cs X
constractor

using System;
class Animal {
    public Animal() {
        Console.WriteLine("constructing....");
    }
}
public class constractor
{
    public static void Main(string[] args)
    {
        Animal obj = new Animal();
        Console.ReadKey();
    }
}
```

file:///c:/users/hp/documents/visual studio 2010/Projects/Cons
constructing....

16. For loop programme:

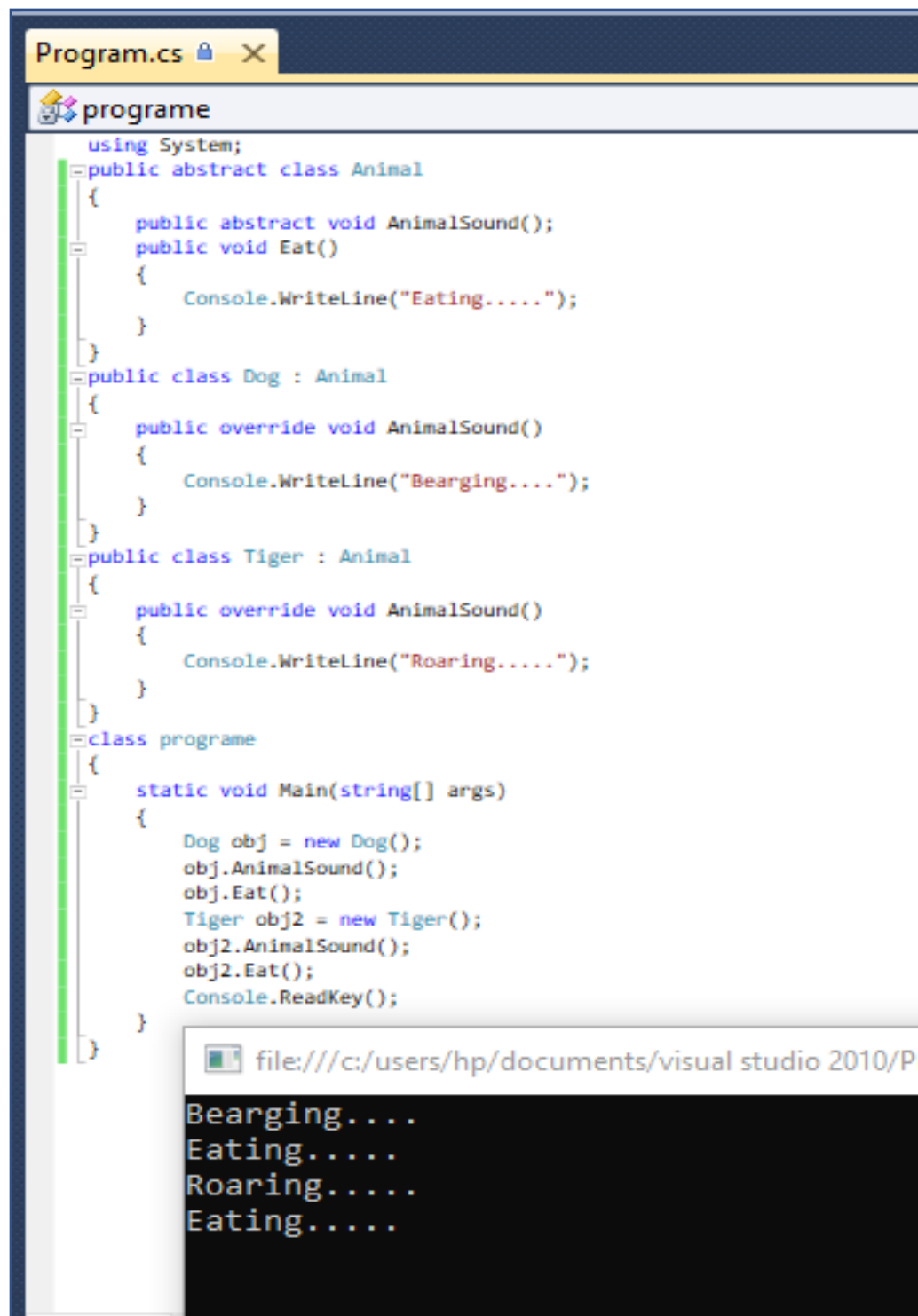


```
Program.cs
programme
using System;
class programme {
    static void Main(string[] args)
    {
        int i, sum = 0;
        for (i = 1; i <= 100; i++)
        {
            sum = sum + i;
        }
        Console.WriteLine("1+2+3.....+100 = {0}", sum);
        Console.ReadKey();
    }
}
```

file:///c:/users/hp/documents/visual studio 2010/Projects/sum between 1 and 100

1+2+3.....+100 = 5050

17. Abstrutor method programe:



The screenshot shows a Visual Studio 2010 IDE window titled "Program.cs". The code defines an abstract class `Animal` with two methods: `AnimalSound()` (abstract) and `Eat()` (concrete). Two subclasses, `Dog` and `Tiger`, inherit from `Animal` and override `AnimalSound()` to output "Bearging...." and "Roaring...." respectively. The `program` class contains a `Main` method that creates instances of `Dog` and `Tiger`, calls their `AnimalSound` and `Eat` methods, and reads a key from the console.

```
using System;
public abstract class Animal
{
    public abstract void AnimalSound();
    public void Eat()
    {
        Console.WriteLine("Eating.....");
    }
}
public class Dog : Animal
{
    public override void AnimalSound()
    {
        Console.WriteLine("Bearging....");
    }
}
public class Tiger : Animal
{
    public override void AnimalSound()
    {
        Console.WriteLine("Roaring.....");
    }
}
class program
{
    static void Main(string[] args)
    {
        Dog obj = new Dog();
        obj.AnimalSound();
        obj.Eat();
        Tiger obj2 = new Tiger();
        obj2.AnimalSound();
        obj2.Eat();
        Console.ReadKey();
    }
}
```

file:///c:/users/hp/documents/visual studio 2010/P

Bearging....
Eating.....
Roaring.....
Eating.....

18. Overloading method programme:

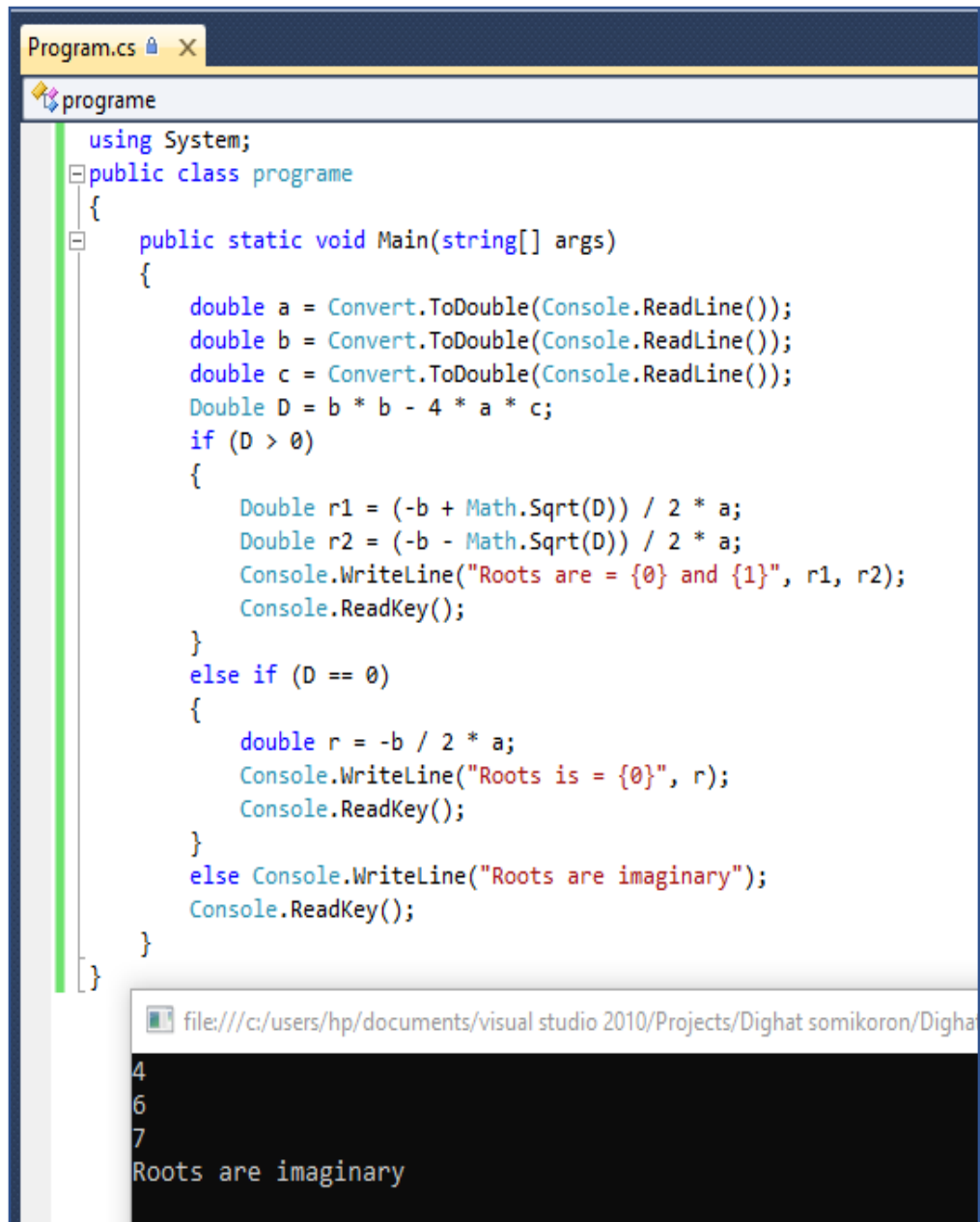


```
Program.cs X
programme
using System;
public class sum
{
    public void Add(int x, int y)
    {
        Console.WriteLine(x + y);
    }
    public void Add(int x, int y, int z)
    {
        Console.WriteLine(x + y + z);
    }
    public void Add(double x, double y)
    {
        Console.WriteLine(x + y);
    }
}
class programme
{
    static void Main(string[] srgs)
    {
        sum obj = new sum();
        obj.Add(10, 20);
        obj.Add(30, 70, 50);
        obj.Add(15.331, 5.285);
        Console.ReadKey();
    }
}
```

file:///c:/users/hp/documents/visual studio 2010/Projects/O

30
150
20.616

19. Dighat somikoron ar Programme:



```
Program.cs
programme
using System;
public class programme
{
    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("Roots are = {0} and {1}", r1, r2);
            Console.ReadKey();
        }
        else if (D == 0)
        {
            double r = -b / 2 * a;
            Console.WriteLine("Roots is = {0}", r);
            Console.ReadKey();
        }
        else Console.WriteLine("Roots are imaginary");
        Console.ReadKey();
    }
}
```

file:///c:/users/hp/documents/visual studio 2010/Projects/Dighat somikoron/Digha

4
6
7
Roots are imaginary