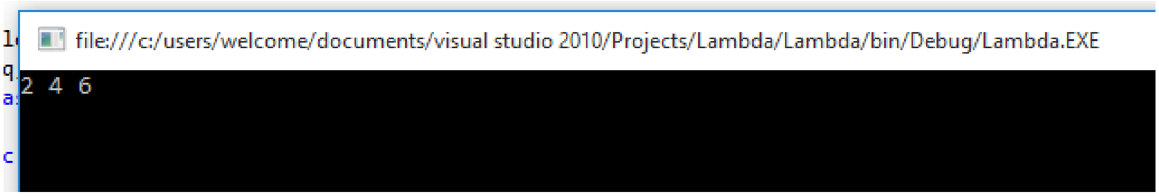
DOT NET TECHNOLOGIES

208W1A12A0 19-10-22 (Lab Task)

Lambda Expressions

```
using System;
using System.Collections.Generic;
using System.Linq;
public static class demo
{
    public static void Main()
    {
        List<int> list = new List<int>() { 1, 2, 3, 4, 5, 6 };
        List<int> evenNumbers = list.FindAll(x => (x % 2) == 0);
        foreach (var num in evenNumbers)
        {
            Console.Write("{0} ", num);
        }
        Console.WriteLine();
        Console.Read()
    }
}
```

OUTPUT:

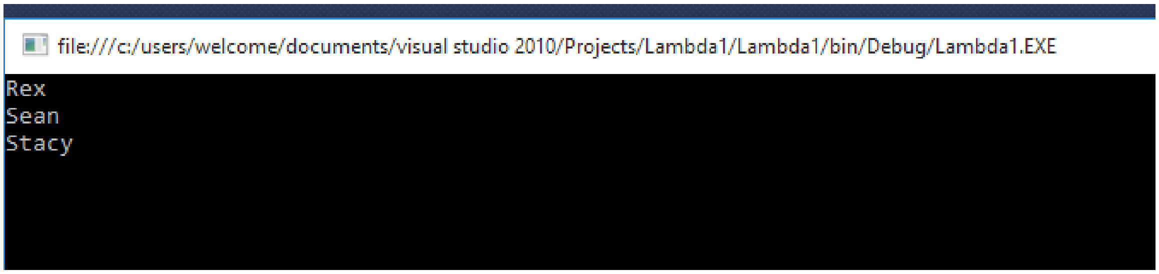


file:///c:/users/welcome/documents/visual studio 2010/Projects/Lambda/Lambda/bin/Debug/Lambda.EXE

```
2 4 6
```

```
using System;

using System.Collections.Generic;

using System.Linq;

class Dog

{

    public string Name { get; set; }

    public int Age { get; set; }

}

class demo

{

    static void Main()

    {

        List<Dog> dogs = new List<Dog>() {

            new Dog { Name = "Rex", Age = 4 },

            new Dog { Name = "Sean", Age = 0 },

            new Dog { Name = "Stacy", Age = 3 }

         };

        var names = dogs.Select(x => x.Name);

        foreach (var name in names)

        {
            Console.WriteLine(name);
        }
        Console.Read();
    }
}
```

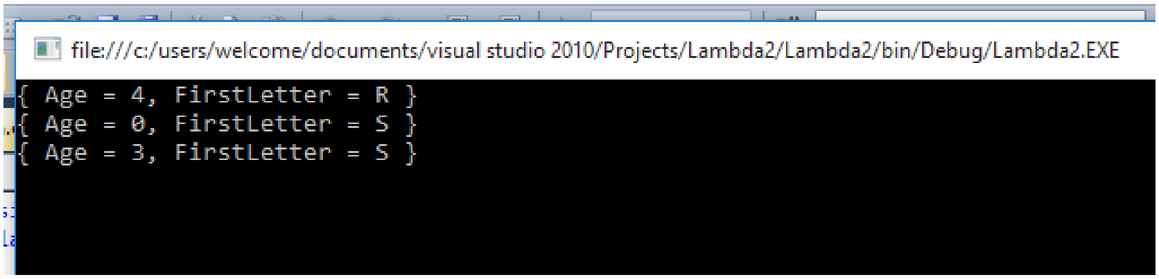
OUTPUT:


file:///c:/users/welcome/documents/visual studio 2010/Projects/Lambda1/Lambda1/bin/Debug/Lambda1.EXE

```
Rex
Sean
Stacy
```

```
using System;
using System.Collections.Generic;
using System.Linq;
class Dog
{
    public string Name { get; set; }
    public int Age { get; set; }
}
class demo
{
    static void Main()
    {
            List<Dog> dogs = new List<Dog>() {
             new Dog { Name = "Rex", Age = 4 },
             new Dog { Name = "Sean", Age = 0 },
             new Dog { Name = "Stacy", Age = 3 }
          };
            var newDogsList = dogs.Select(x => new { Age = x.Age, FirstLetter = x.Name[0] });
            foreach (var item in newDogsList)
            {
                Console.WriteLine(item);
            }
            Console.Read();
        }
    }
```

OUTPUT:

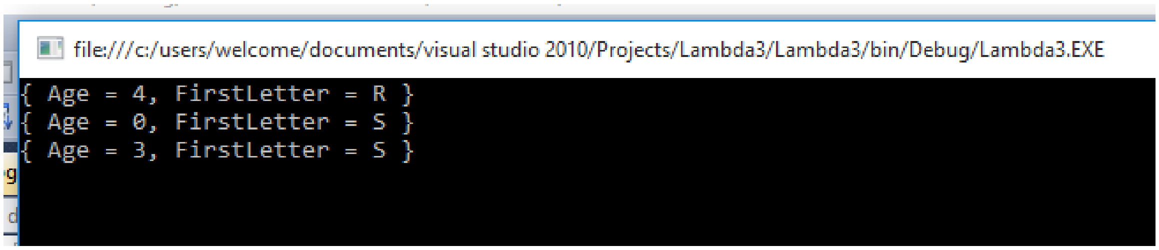


file:///c:/users/welcome/documents/visual studio 2010/Projects/Lambda2/Lambda2/bin/Debug/Lambda2.EXE

```
{ Age = 4, FirstLetter = R }
{ Age = 0, FirstLetter = S }
{ Age = 3, FirstLetter = S }
```

```
using System;
using System.Collections.Generic;
using System.Linq;
class Dog
{
    public string Name { get; set; }
    public int Age { get; set; }
}
class demo
{
    static void Main()
    {
        List<Dog> dogs = new List<Dog>() {
         new Dog { Name = "Rex", Age = 4 },
         new Dog { Name = "Sean", Age = 0 },
         new Dog { Name = "Stacy", Age = 3 }
      };
        var newDogsList = dogs.Select(x => new { Age = x.Age, FirstLetter = x.Name[0] });
        foreach (var item in newDogsList)
        {
            Console.WriteLine(item);
        }
        Console.Read();
    }
}
```

OUTPUT:

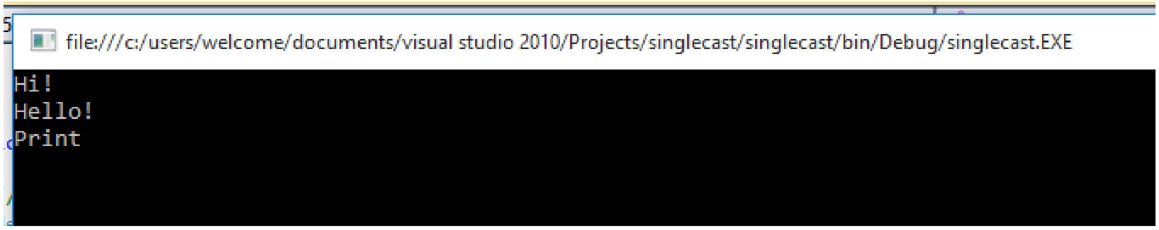

file:///c:/users/welcome/documents/visual studio 2010/Projects/Lambda3/Lambda3/bin/Debug/Lambda3.EXE

```
{ Age = 4, FirstLetter = R }
{ Age = 0, FirstLetter = S }
{ Age = 3, FirstLetter = S }
```

DELEGATES

```
using System;
namespace ConsoleApplication5
{
    class Program
    {
        public delegate void delmethod();
        public class P
        {
            public static void display()
            {
                Console.WriteLine("Hello!");
            }
            public static void show()
            {
                Console.WriteLine("Hi!");
            }
            public void print()
            {
                Console.WriteLine("Print");
            }
        }

        static void Main(string[] args)
        {
            // here we have assigned static method show() of class P to delegate delmethod()
            delmethod del1 = P.show;

            // here we have assigned static method display() of class P to delegate delmethod() using new operator
            // you can use both ways to assign the delagate
            delmethod del2 = new delmethod(P.display);
            P obj = new P();

            // here first we have create instance of class P and assigned the method print() to the delegate i.e. delegate with class
            delmethod del3 = obj.print;
            del1();
            del2();
            del3();
            Console.ReadLine();
        }
    }
}
```

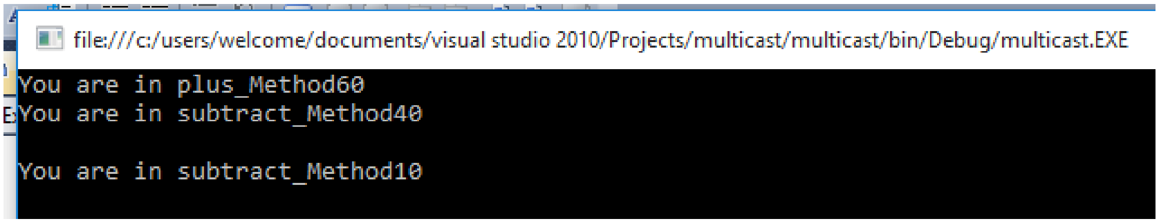
OUTPUT:


file:///c:/users/welcome/documents/visual studio 2010/Projects/singlecast/singlecast/bin/Debug/singlecast.EXE

```
Hi!
Hello!
Print
```

```
using System;
namespace delegate_Example4
{
    class Program
    {
        public delegate void delmethod(int x, int y);

        public class TestMultipleDelegate
        {
            public void plus_Method1(int x, int y)
            {
                Console.Write("You are in plus_Method");
                Console.WriteLine(x + y);
            }
            public void subtract_Method2(int x, int y)
            {
                Console.Write("You are in subtract_Method");
                Console.WriteLine(x - y);
            }
        }
        static void Main(string[] args)
        {

            TestMultipleDelegate obj = new TestMultipleDelegate();
            delmethod del = new delmethod(obj.plus_Method1);

            // Here we have multicast
            del += new delmethod(obj.subtract_Method2);
            // plus_Method1 and subtract_Method2 are called
            del(50, 10);
            Console.WriteLine();
            //Here again we have multicast
            del -= new delmethod(obj.plus_Method1);
            //Only subtract_Method2 is called
            del(20, 10);
            Console.ReadLine();
        }
    }
}
```

OUTPUT:



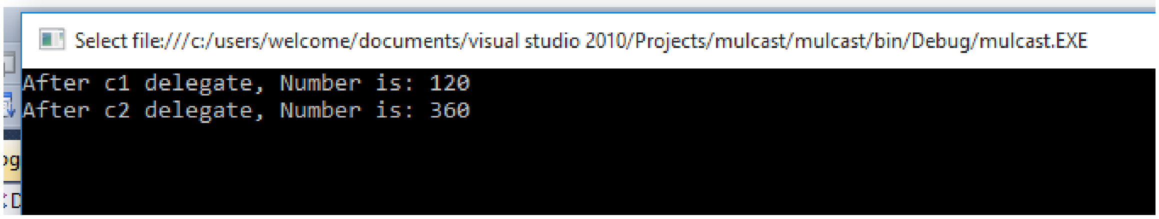
file:///c:/users/welcome/documents/visual studio 2010/Projects/multicast/multicast/bin/Debug/multicast.EXE

```
You are in plus_Method60
You are in subtract_Method40

You are in subtract_Method10
```

```
using System;
delegate int Calculator(int n);//declaring delegate
public class DelegateExample
{
    static int number = 100;
    public static int add(int n)
    {
        number = number + n;
        return number;
    }
    public static int mul(int n)
    {
        number = number * n;
        return number;
    }
    public static int getNumber()
    {
        return number;
    }
    public static void Main(string[] args)
```

```
        {
            Calculator c1 = new Calculator(add);//instantiating delegate
            Calculator c2 = new Calculator(mul);
            c1(20);//calling method using delegate
            Console.WriteLine("After c1 delegate, Number is: " + getNumber());
            c2(3);
            Console.WriteLine("After c2 delegate, Number is: " + getNumber());
            Console.ReadLine();
        }
    }
```

OUTPUT:



Select file:///c:/users/welcome/documents/visual studio 2010/Projects/mulcast/mulcast/bin/Debug/mulcast.EXE

```
After c1 delegate, Number is: 120
After c2 delegate, Number is: 360
```