**Lab2**

**Q1.**

**Code(ConsoleApp)**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ItemTrigger

{

class Item

{

public delegate void priceChangedEvenHandler(int price);

public event priceChangedEvenHandler PriceChanged;

private int price;

private string name;

public int Price

{

get { return price; }

set

{

if (value > 0)

{

price = value;

if (PriceChanged != null)

PriceChanged(value);

}

}

}

public string Name { get; set; }

}

class Q1

{

private static void printNewPrice(int price)

{

Console.WriteLine("[Event Handler] Price Changed to : " + price);

}

static void Main()

{

int price;

Console.Write("Enter Price: ");

Item p = new Item();

p.PriceChanged += printNewPrice;

int.TryParse(Console.ReadLine(), out price);

p.Price = price;

Console.WriteLine("[Main] Price updated");

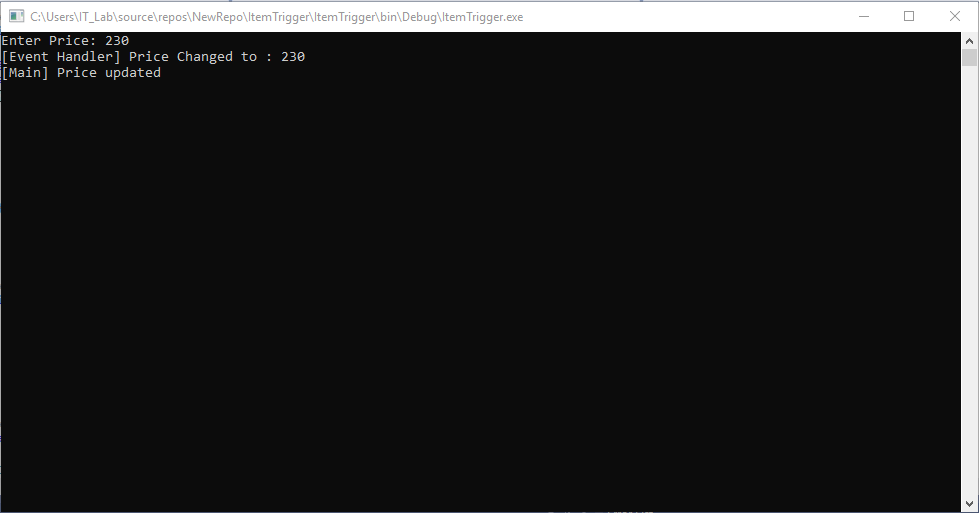
Console.Read();

}

}

}

**Output**

****

**Q2.**

**Code(ConsoleApp)**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace GstCal

{

class Item

{

public string Name { get; set; }

private int cost;

public int Cost

{

get { return cost; }

set

{

if (value > 0)

cost = value;

}

}

public double GST()

{

return 0.08 \* cost;

}

}

class Program

{

static void Main(string[] args)

{

Item i = new Item();

int cost;

Console.WriteLine("Enter Item name");

i.Name = Console.ReadLine();

Console.WriteLine("Enter the cost");

int.TryParse(Console.ReadLine(), out cost);

i.Cost = cost;

Console.WriteLine("\nItem Name: " + i.Name +

"\nCost: " + i.Cost +

"\nGST(8%): " + i.GST() +

"\n\nTotal: " + (i.Cost + i.GST()));

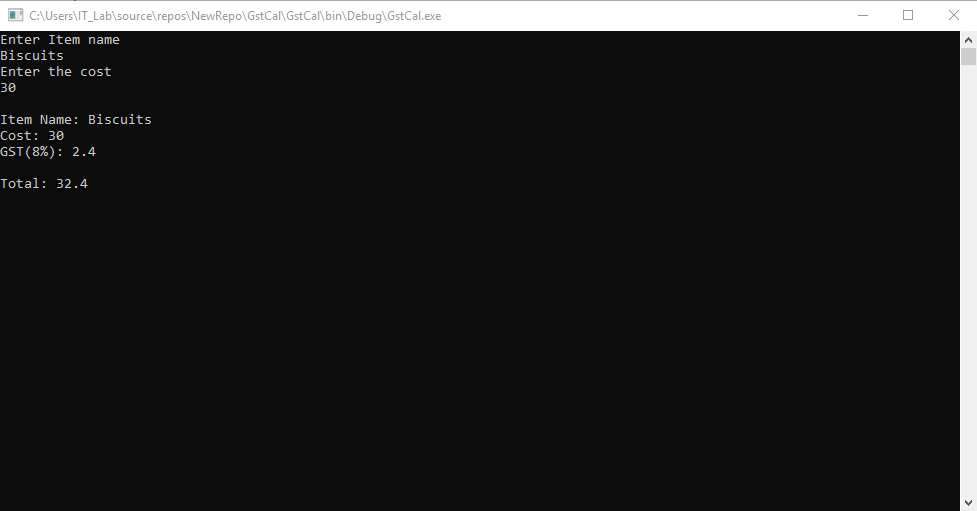
Console.Read();

}

}

}

**Output**



**Q3.**

**Code(ConsoleApp)**

using System;

namespace TrafficDelegate

{

public delegate string TrafficDel();

class Program

{

static void Main(string[] args)

{

TrafficDel tDel;

tDel = Red;

//string temp = "";

string curr\_colour = tDel();

Console.WriteLine("Signal is " + curr\_colour);

tDel = Yellow;

curr\_colour = tDel();

Console.WriteLine("Signal is " + curr\_colour);

tDel = Green;

curr\_colour = tDel();

Console.WriteLine("Signal is " + curr\_colour);

Console.ReadLine();

}

private static string Yellow()

{

return "Yellow";

}

private static string Red()

{

return "Red";

}

private static string Green()

{

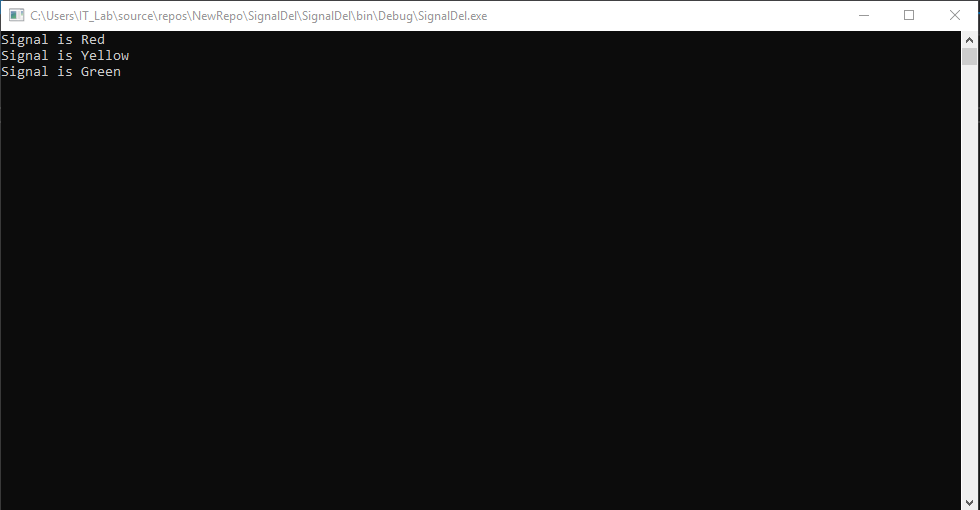
return "Green";

}

}

}

**Output**

****