1. **Create a class called "Item" which has the following properties: a. Name of the item b. Price of the item Create an event in Item class that is fired whenever price changes through the property created. Create a console application that will display the new price of the Item using an EventHandler.**

Code:

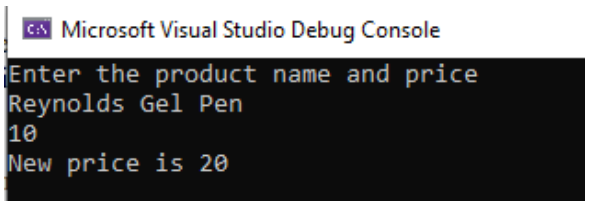
```
using System;

namespace q1
{
    public delegate void PriceChangedEventHandler(double price);
    class Item
    {
        private string name;
        private double price;
        public event PriceChangedEventHandler PriceChanged;

        public string Name { get; set; }
        public double Price {
            get
            {
                return price;
            }
            set
            {
                price = value;
                if (PriceChanged != null)
                    PriceChanged(value);
            }
        }

    }
    class Program
    {
        static void Main(string[] args)
        {
            Item item1 = new Item();
            Console.WriteLine("Enter the product name and price");
            item1.Name = Console.ReadLine();
            double price;
            double.TryParse(Console.ReadLine(), out price);
            item1.Price = price;
            item1.PriceChanged += ChangeDetected; //Linking Event Handler to Event
            item1.Price = item1.Price * 2;
        }
        //Event handler
        public static void ChangeDetected(double price)
        {
            Console.WriteLine("New price is " + price);
        }
    }
}
```

Output:



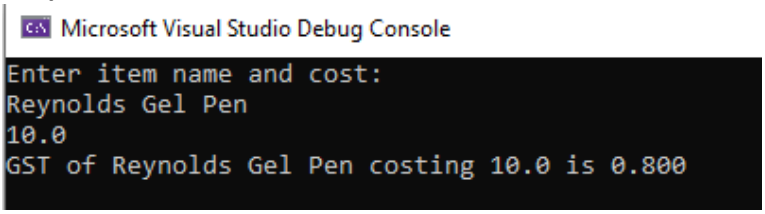
2. **Create a console application with class named "Item" which contains one automatic property called "name" of type string and a shared property called "cost" which takes only positive decimal values. Write a function "CalcGst( )" which returns a decimal value (Formula for GST:8% of the base cost).**

Code:

```
using System;

namespace q2
{
    class Item
    {
        private static decimal cost;
        public static decimal Cost
        {
            get
            {
                return cost;
            }
            set
            {
                //Ensure it is a positive decimal
                if (value < 0)
                {
                    Console.WriteLine("Invalid. Only positive decimals accepted.");
                }
                else
                    cost = value;
            }
        }
        public string Name { get; set; }
        public decimal CalcGst()
        {
            return 0.08M * cost;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Item item = new Item();
            Console.WriteLine("Enter item name and cost: ");
            item.Name = Console.ReadLine();
            decimal cost;
            //Check if input is decimal
            Decimal.TryParse(Console.ReadLine(), out cost);
            Item.Cost = cost;
            Console.WriteLine("GST of {0} costing {1} is {2} ", item.Name, Item.Cost,
       item.CalcGst());
        }
    }
}
```

Output:



3. **Write a program in C# to demonstrate declaration, instantiation, and use of a delegate, called TrafficDel and a class called TrafficSignal with the following delegate methods Yellow() ,Green() and Red() which prints what each signal is meant for.**

Code:

```
using System;

namespace q3
{
    public delegate void TrafficDel();
    class TrafficSignal
    {
        public void Yellow()
        {
            Console.WriteLine("Yellow");
        }
        public void Green()
        {
            Console.WriteLine("Green");
        }
        public void Red()
        {
            Console.WriteLine("Red");
        }

    }
    class Program
    {
        static void Main(string[] args)
        {
            //TrafficSignal class object
            TrafficSignal trafficSignal = new TrafficSignal();

            //Create and assign the delegate to Red delegate method and then call it
            TrafficDel signalDelegate = trafficSignal.Red;
            signalDelegate();

            //Assign the delegate to Yellow delegate method and then call it
            signalDelegate = trafficSignal.Yellow;
            signalDelegate();

            //Assign the delegate to Green delegate method and then call it
            signalDelegate = trafficSignal.Green;
            signalDelegate();


        }
    }
}
```

Output: