```
//stock.cs
using System.Collections.Generic;
using System.Text;
using System.Threading;

namespace Stock
{
//---------------------------------------------------------------------------------

    public class Stock
    {
        public event EventHandler<StockNotification> StockEvent;


        //Name of our stock.
        private string _name;
        //Starting value of the stock.
        private int _initialValue;
        //Max change of the stock that is possible.
        private int _maxChange;
        //Threshold value where we notify subscribers to the event.
        private int _threshold;
        //Amount of changes the stock goes through.
        private int _numChanges;
        //Current value of the stock.
        private int _currentValue;


        private readonly Thread _thread;
        public string StockName { get => _name; set => _name = value; }
        public int InitialValue
        public int CurrentValue
        public int MaxChange
        public int Threshold
        public int NumChanges
        //---------------------------------------------------------------------------

        /// <summary>
        /// Stock class that contains all the information and changes of the stock
        /// </summary>
        /// <param name="name">Stock name</param>
        /// <param name="startingValue">Starting stock value</param>
        /// <param name="maxChange">The max value change of the stock</param>
        /// <param name="threshold">The range for the stock</param>
        public Stock(string name, int startingValue, int maxChange, int threshold)
        {
            _name = name;
            _initialValue = startingValue;
            _currentValue = InitialValue;
            _maxChange = maxChange;
            _threshold = threshold;
            this._thread = new Thread(new ThreadStart(_____________________));
            _thread._________________;
        }

        //---------------------------------------------------------------------------
----
```

```
        /// <summary>
        /// Activates the threads synchronizations
        /// </summary>
        public void Activate()
        {
            for (int i = 0; i < 25; i++)
            {
                Thread.Sleep(500); // 1/2 second
                ChangeStockValue();
            }
        }

//-------------------------------------------------------------------------------------

        // delegate
        //public delegate void StockNotification(String stockName, int currentValue, int
numberChanges);

        // event
        //public event StockNotification ProcessComplete;
        //-------------------------------------------------------------------------------
-----------

        /// <summary>
        /// Changes the stock value and also raising the event of stock value changes
        /// </summary>
        public void ChangeStockValue()
        {
            var rand = new Random();
            CurrentValue += rand.Next(1, MaxChange);
            NumChanges++;
            if ((CurrentValue - InitialValue) > Threshold)
            {  //RAISE THE EVENT
                      _____ Invoke ____________________________________________________


            }
        }
//-------------------------------------------------------------------------------------
---------
    }
}
```

```
//stockbroker.cs



using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
using System.Threading;

namespace Stock
{
    //!NOTE!: Class StockBroker has fields broker name and a list of Stock named stocks.
    //       addStock method registers the Notify listener with the stock (in addition to
```

```
    //       adding it to the lsit of stocks held by the broker). This notify method
outputs
    //       to the console the name, value, and the number of changes of the stock whose
    //       value is out of the range given the stock's notification threshold.
    public class StockBroker
    {
        public string BrokerName { get; set; }

        public List<Stock> stocks = new List<Stock>();

        public static ReaderWriterLockSlim myLock = new ReaderWriterLockSlim();
        //readonly string docPath = @"C:\Users\Documents\CECS 475\Lab3_output.txt";

        readonly string destPath = Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"Lab1_output.txt");

        public string titles = "Broker".PadRight(10) + "Stock".PadRight(15) +
            "Value".PadRight(10) + "Changes".PadRight(10) + "Date and Time";
//-------------------------------------------------------------------------------------
--------------------------

        /// <summary>
        ///     The stockbroker object
        /// </summary>
        /// <param name="brokerName">The stockbroker's name</param>
        public StockBroker(string brokerName)
        {
            BrokerName = brokerName;
        }
//-------------------------------------------------------------------------------------
--------------------------

        /// <summary>
        ///     Adds stock objects to the stock list
        /// </summary>
        /// <param name="stock">Stock object</param>
        public void AddStock(Stock stock)
        {
            stocks._____________________________
            stock._____________________________________
        }
//-------------------------------------------------------------------------------------
------------------------------

        /// <summary>
        ///     The eventhandler that raises the event of a change
        /// </summary>
        /// <param name="sender">The sender that indicated a change</param>
        /// <param name="e">Event arguments</param>
        void EventHandler(Object sender, EventArgs e)
        {
            try
            {    //LOCK Mechanism
                 _____________________________
                Stock newStock = (Stock)sender;
                //string statement;
                //!NOTE!: Check out C#events, pg.4
                // Display the output to the console windows
```

```
                Console.WriteLine(BrokerName.PadRight(16)
_____________________________________________);
                //Display the output to the file
                using (StreamWriter outputFile
=_____________________________________________)
                {
                    _____________________________________________
                }
                //RELEASE the lock
                ____________________
            }
            finally
            {

            }

        }
//-------------------------------------------------------------------------------------------
------------------------------
    }
}
```

Stocknotification.cs

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Stock
{
    public class StockNotification : EventArgs
    {
        public string StockName { get; set; }
        public int CurrentValue { get; set; }
        public int NumChanges { get; set; }

        /// <summary>
        /// Stock notification attributes that are set and changed
        /// </summary>
        /// <param name="stockName">Name of stock</param>
        /// <param name="currentValue">Current vallue of the stock</param>
        /// <param name="numChanges">Number of changes the stock goes through</param>
        public StockNotification(string stockName, int currentValue, int numChanges)
        {
            // !NOTE!: Fill in below of what the notification will do using the comments
above
            this.StockName = stockName;
            this.CurrentValue = currentValue;
            this.NumChanges = numChanges;
        }
    }
}
```

Extras:

```
string titles = "Broker".PadRight(16) + "Stock".PadRight(16) +
            "Value".PadRight(16) + "Changes".PadRight(10) + "Date and Time";

            Console.WriteLine(titles);

            string destPath = Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"Lab1_output.txt");
            using (StreamWriter outputFile = new StreamWriter(destPath, false))
```