Lab#3 – Asynchronous Programming & Parallel Programming

Due Date: Friday Midnight of Week#6

Purpose: The purpose of this assignment is to help you become familiar with

- Asynchronous programming & Parallel programming
- WPF

Instructions: Be sure to read the following general instructions carefully:

> This lab must be completed individually. Please submit your solution **through the dropbox**, and name your submission according to the following pattern: **studentID(yourlastname)_Labnumber.zip**. e.g., 300123456(**smith)_LAB#3**.zip

Rubric

	Functionality	Marks
Q1	1. GUI	2
	2. Class that is to model stock data	1
	3. Load data from csv file and process data (use asynchronous programming) 3.1.1 Read (1 mark) 3.1.2 Clean data (1 mark) 3.1.3 Display loaded data in DataGrid (2.5 mark) 3.1.4 Sort data according to date (0.5 mark)	5
	4. Calculate factorial number asynchronously	1
	5. Overall (i.e., Usability and readability, etc.)	1
Q2	1. Use ***parallel.Invoke()*** to	1
	1.1. Display selected image	1
	1.2. Save selected image locally	1
	1.3. Create and save thumbnail locally	1
	2. Overall	1

Question 1[10 marks]

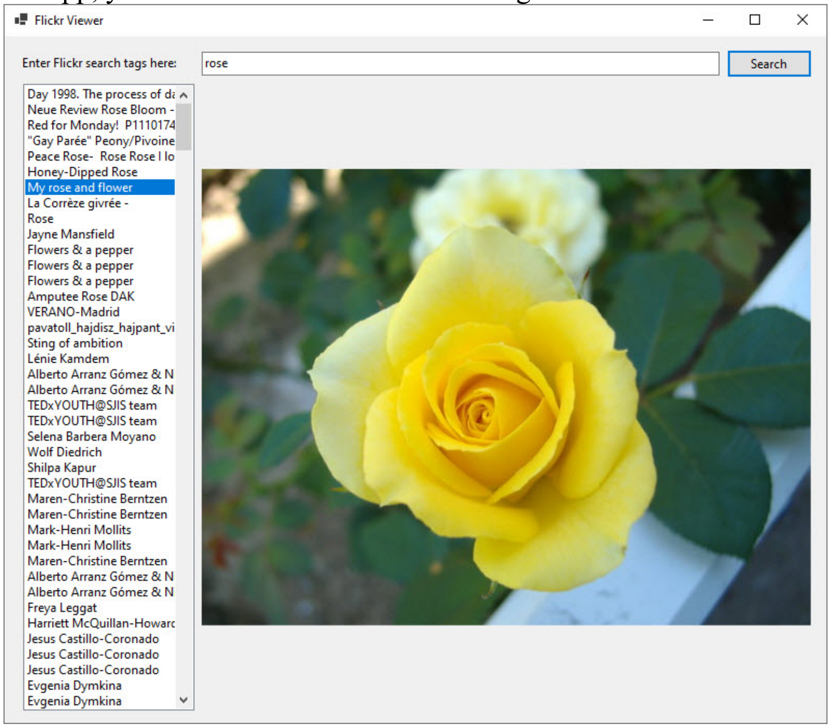
Implement a **WPF** app to list stock information that is loaded from ***StockData.csv***. Your app should allow user to ***Search*** specific company's stock information, and display the search result in a ***DataGrid***, and the result should be sorted according to the date. Clean the data by removing those rows that the price contains negative value while the data is loading.

Your app should also allow user to calculate factorial number. The factorial number calculation should not block user to interact with your app.
[Hint: File I/O and calculating factorial number are typical scenario to use asynchronous programming]

Question 2[5 marks]

FlickrViewer app allows user to search photo(s) based on tag from www.flickr.com through Flickr API. A tag is any user-supplied word or phrase that helps organize web content. In order to run the application, you need to apply an API key from https://www.flickr.com/services/apps/create/apply.

When you launch the app, you should be able to see something similar as below.

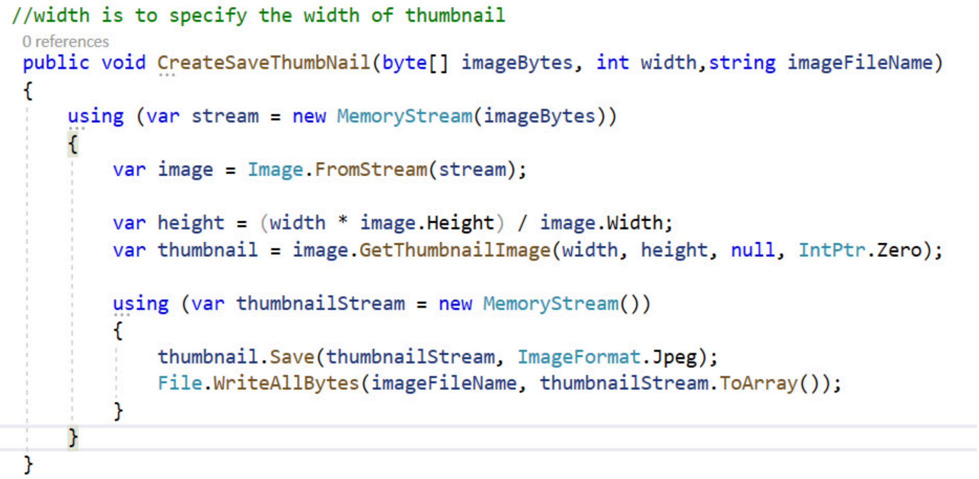


Day 1998. The process of da
Neue Review Rose Bloom -
Red for Monday! P1110174
"Gay Parée" Peony/Pivoine
Peace Rose- Rose Rose I lo
Honey-Dipped Rose
My rose and flower
La Corrèze givrée -
Rose
Jayne Mansfield
Flowers & a pepper
Flowers & a pepper
Flowers & a pepper
Amputee Rose DAK
VERANO-Madrid
pavatoll_hajdisz_hajpant_vi
Sting of ambition
Lénie Kamdem
Alberto Arranz Gómez & N
Alberto Arranz Gómez & N
TEDxYOUTH@SJIS team
TEDxYOUTH@SJIS team
Selena Barbera Moyano
Wolf Diedrich
Shilpa Kapur
TEDxYOUTH@SJIS team
Maren-Christine Berntzen
Maren-Christine Berntzen
Mark-Henri Mollits
Mark-Henri Mollits
Maren-Christine Berntzen
Alberto Arranz Gómez & N
Alberto Arranz Gómez & N
Freya Leggat
Harriett McQuillan-Howarc
Jesus Castillo-Coronado
Jesus Castillo-Coronado
Jesus Castillo-Coronado
Evgenia Dymkina
Evgenia Dymkina

You are asked to modify ***FlickrViewer*** app to allow user save selected image locally, create and save thumbnail locally. More specifically, after user select one item from the listBox (located at left), your app should perform following three actions parallelly by using ***Parallel*** class:

1) display the selected image in the PictureBox (located at right), which the original ***FlickrViewer*** app has already provided
2) Save the selected image locally
3) Create thumbnail for the selected image, and save it locally

[***Hint:*** following snippet shows you how to create and save thumbnail]



```
//width is to specify the width of thumbnail
 0 references
 public void CreateSaveThumbNail(byte[] imageBytes, int width,string imageFileName)
 {
     using (var stream = new MemoryStream(imageBytes))
     {
         var image = Image.FromStream(stream);

         var height = (width * image.Height) / image.Width;
         var thumbnail = image.GetThumbnailImage(width, height, null, IntPtr.Zero);

         using (var thumbnailStream = new MemoryStream())
         {
             thumbnail.Save(thumbnailStream, ImageFormat.Jpeg);
             File.WriteAllBytes(imageFileName, thumbnailStream.ToArray());
         }
     }
 }
```

Reference: https://www.andrewhoefling.com/Home/post/basic-image-manipulation-in-c-sharp