Capstone Assignment

October 28, 2019

In [1]:
```
import zipfile
from zipfile import ZipFile

from PIL import Image, ImageDraw
import pytesseract
import cv2 as cv
import numpy as np
from kraken import pageseg

# loading the face detection classifier
face_cascade = cv.CascadeClassifier('readonly/haarcascade_frontalface_default.xml')

pages = []



def crop_thumbnails(list1, image):
    cropped_faces = []
    for item in list1:
        img = image.copy()
        cropped_face = img.crop((item[0], item[1], item[0]+item[2], item[1]+item[3]))
        cropped_face.thumbnail((100,100))
        cropped_faces.append(cropped_face)
    return cropped_faces


def create_contact_sheet(list1):
    first_image=list1[0]
    if len(list1) <= 5:
        contact_sheet= Image.new(first_image.mode, (100*5,100), (0,0,0))
    elif len(list1) > 5 and len(list1) <= 10:
        contact_sheet= Image.new(first_image.mode, (100*5,100*2), (0,0,0))
    x=0
    y=0
    for item in list1:
        contact_sheet.paste(item, (x, y) )
        if x+100 == contact_sheet.width:
```

```
            x=0
            y=y+100
        else:
            x=x+100
    return contact_sheet



with ZipFile('readonly/small_img.zip', 'r') as zip:
    zip_lst = ZipFile.infolist(zip)
    for z in zip_lst:
        with zip.open(z) as fhand:
            img = Image.open(fhand)
            img_gray = img.convert('L')
            try:
                text = pytesseract.image_to_string(img_gray)
            except:
                pass
            if 'Christopher' in text:
                img_cv = np.asarray(img)
                img_cv_gray = cv.cvtColor(img_cv, cv.COLOR_BGR2GRAY)
                faces = face_cascade.detectMultiScale(img_cv_gray, 1.3, 5)
                dict = {}
                dict['image'] = img
                dict['filename'] = z.filename
                try:
                    dict['faces'] = faces.tolist()
                except:
                    dict['faces'] = []
                pages.append(dict)

with ZipFile('readonly/images.zip', 'r') as zip:
    zip_lst = ZipFile.infolist(zip)
    for z in zip_lst:
        with zip.open(z) as fhand:
            img = Image.open(fhand)
            img_gray = img.convert('L')
            try:
                text = pytesseract.image_to_string(img_gray)
            except:
                pass
            if 'Mark' in text:
                img_cv = np.asarray(img)
                img_cv_gray = cv.cvtColor(img_cv, cv.COLOR_BGR2GRAY)
                faces = face_cascade.detectMultiScale(img_cv_gray, 1.3, 5)
                dict = {}
                dict['image'] = img
                dict['filename'] = z.filename
```

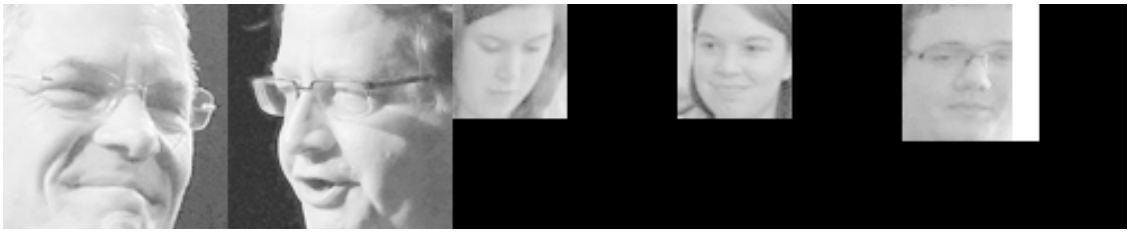
```
                        try:
                            dict['faces'] = faces.tolist()
                        except:
                            dict['faces'] = []
                        pages.append(dict)

    for page in pages:
        if len(page['faces']) > 0:
            print("Results found in file {}".format(page['filename']))
            display(create_contact_sheet(crop_thumbnails(page['faces'], page['image'])))
        elif len(page['faces']) == 0:
            print("Results found in file {}".format(page['filename']))
            print("But there were no faces in that file!")
```

```
Results found in file a-0.png
```

```
Results found in file a-3.png
```

```
Results found in file a-0.png
```

```
Results found in file a-1.png
```



```
Results found in file a-10.png
But there were no faces in that file!
Results found in file a-13.png
```

```
Results found in file a-2.png
```

```
Results found in file a-3.png
```

```
Results found in file a-8.png
But there were no faces in that file!
```