In [1]:
```
# Question 1 - Import data
import pandas as pd
import numpy as np
import os

path = "D:\\Documents\\DAAN862\\"
os.chdir(path)
file_reg = "Registration.csv"
file_course = "Course_info.xlsx"

reg_pd = pd.read_csv(file_reg)
course_pd = pd.read_excel(file_course)

# Question 2 - Explore and Clean Registration
# We will create a new data frame containing any rows
# with identified nulls to see how/what needs handled
reg_pd
# As shown, the registration file contains 4900 rows
# of various named students with each row corresponding
# to a semester date and coursename
```

Out[1]:

	Student name	semester new	coursename
0	Bill Mumy	Fall 2004	BEHAVIORAL PHARMACOLOGY
1	Bill Mumy	Fall 2000	AMERICAN FOREIGN POLICY
2	Bill Mumy	Fall 2003	DRUGS, BRAIN AND MIND
3	Bill Mumy	Fall 2005	Environmental Case Studies
4	Bill Mumy	Fall 2000	COMPUTER LINEAR ALGEBRA
...	...	...	...
4895	Stacy Keach	Summer 2001	CELL. BIOL. And BIOCHEM.
4896	Ann Landers	Summer 2004	AMERICAN HEALT POLICY
4897	Ann Landers	Summer 2004	ANALYTICAL MECHANICS
4898	Tyne Daly	Summer 2004	COMPUT LINEAR ALGEBRA
4899	Tyne Daly	Summer 2004	EXPERIMENTAL WRITING SEM: The Ecology of Poetry

4900 rows × 3 columns

In [2]:
```
# Some potentially useful information as well:
# Number of unique students and classes registered
unique_students = reg_pd['Student name'].unique()
unique_courses = reg_pd['coursename'].unique()

print("Unique Number of Students: " + str(len(unique_students)))
print("Unique Number of Course: " + str(len(unique_courses)))
```

```
Unique Number of Students: 448
Unique Number of Course: 169
```

In [3]:
```
# We should also look for duplicates within registration,
# so we will pull from a few names to see if we can find any
display(reg_pd[reg_pd['Student name'].str.contains('Bill Mumy')])
```

	Student name	semester new	coursename
0	Bill Mumy	Fall 2004	BEHAVIORAL PHARMACOLOGY
1	Bill Mumy	Fall 2000	AMERICAN FOREIGN POLICY
2	Bill Mumy	Fall 2003	DRUGS, BRAIN AND MIND
3	Bill Mumy	Fall 2005	Environmental Case Studies
4	Bill Mumy	Fall 2000	COMPUTER LINEAR ALGEBRA
5	Bill Mumy	Spring 2002	ART, from ancient to 1945
6	Bill Mumy	Spring 2003	CONTEMP ART - since 1945
7	Bill Mumy	Spring 2003	CONTEMP ART - since 1945
8	Bill Mumy	Spring 2003	CONTEMP ART - since 1945
9	Bill Mumy	Fall 2002	ANALYTICAL MECHANICS
10	Bill Mumy	Fall 2002	ANALYTICAL MECHANICS
11	Bill Mumy	Fall 2002	ANALYTICAL MECHANICS
12	Bill Mumy	Spring 2005	CONTEMPORARY AFRICAN ART
13	Bill Mumy	Spring 2005	CONTEMPORARY AFRICAN ART
14	Bill Mumy	Spring 2005	CONTEMPORARY AFRICAN ART
15	Bill Mumy	Fall 2004	CEL BIO BIOCHEMISTRY
16	Bill Mumy	Fall 2004	CEL BIO BIOCHEMISTRY
17	Bill Mumy	Fall 2004	CEL BIO BIOCHEMISTRY
18	Bill Mumy	Spring 2003	CELL BIOLOGY and BIOCHEMISTRY
19	Bill Mumy	Spring 2003	CELL BIOLOGY and BIOCHEMISTRY
20	Bill Mumy	Spring 2003	CELL BIOLOGY and BIOCHEMISTRY
21	Bill Mumy	Spring 2002	BIBLE IN TRANSLATION: Proverbs, Ecclesiastes, ...
22	Bill Mumy	Spring 2002	BIBLE IN TRANSLATION: Proverbs, Ecclesiastes, ...
23	Bill Mumy	Spring 2002	BIBLE IN TRANSLATION: Proverbs, Ecclesiastes, ...

In [5]:
```
# We can see that there are in fact duplicates in the entries
# Create a new table with duplicates removes
reg_pd_dups = reg_pd.drop_duplicates()
display(reg_pd_dups[reg_pd_dups['Student name'].str.contains('Bill Mumy')])
```

	Student name	semester new	coursename
0	Bill Mumy	Fall 2004	BEHAVIORAL PHARMACOLOGY
1	Bill Mumy	Fall 2000	AMERICAN FOREIGN POLICY
2	Bill Mumy	Fall 2003	DRUGS, BRAIN AND MIND
3	Bill Mumy	Fall 2005	Environmental Case Studies
4	Bill Mumy	Fall 2000	COMPUTER LINEAR ALGEBRA
5	Bill Mumy	Spring 2002	ART, from ancient to 1945
6	Bill Mumy	Spring 2003	CONTEMP ART - since 1945
9	Bill Mumy	Fall 2002	ANALYTICAL MECHANICS
12	Bill Mumy	Spring 2005	CONTEMPORARY AFRICAN ART
15	Bill Mumy	Fall 2004	CEL BIO BIOCHEMISTRY
18	Bill Mumy	Spring 2003	CELL BIOLOGY and BIOCHEMISTRY
21	Bill Mumy	Spring 2002	BIBLE IN TRANSLATION: Proverbs, Ecclesiastes, ...

In [6]:
```
# Obtain a list of nulls found in the data set
reg_nulls = reg_pd_dups[reg_pd_dups.isnull().any(axis = 1)]
display(reg_nulls)
```

	Student name	semester new	coursename
1650	John Jakes	Summer 2003	NaN

In [8]:
```
# As identified, only one row has a missing
# value, which is row 1650 with a missing
# coursename. In this case, we should just
# drop this row in the original data frame
reg_pd_clean = reg_pd_dups.dropna(how = 'any')
display(reg_pd_clean.iloc[1648:1652])

# Before going further, we should also take a
# look at some unique values to see if any
# accidental duplicates exist especially in
# regards to the coursenames
unique_courses = reg_pd_clean['coursename'].unique()
display(sorted(unique_courses))

# With some skimming, its apparent that a lot
# of classes have similar names
# and we may want to rename them to a common name
# Before doing so, we should analyze the courses
# listed in the excel sheet to determine what that
# common name should be
```

```
 'ART - ancient to 1945',
 'ART - from ancient to 1945',
 'ART AND BUSINESS OF FILM',
 'ART AND RELIGION',
 'ART ancient to 1945',
 'ART, from ancient to 1945',
 'ART: ancient to 1945',
 'ASIAN AMER COMM FLD WRK',
 'AUGUSTAN CULTRL REVOL',
 'BECOMING HUMAN',
 'BEG RDG/WRTG CHINESE II',
 'BEHAVIORAL ECON & PSYCH',
 'BEHAVIORAL PHARMACOLOGY',
 'BEING HUMAN: Being Human: Biology, Culture & Human Diversity',
 'BIBLE IN TRANSLATION: Proverbs, Ecclesiastes, and Job',
 'BIOCHEMISTRY RESEARCH',
 'BIOLOGICAL CHEMISTRY II',
 'BRITAIN SINCE 1945',
 'BRITISH POETRY 1660-1914',
 'Business German - Micro Perspective',
```

In [9]:
```
# Question 3 - Explore and Clean Courses
# We will use the same tactic form Question 2
display(course_pd)

# Obtain any nulls
course_nulls = course_pd[course_pd.isnull().any(axis = 1)]
display(course_nulls)
# Since we have identified that a course exists
# without a name, this may be the missing coursename
# from the registration file. At this point, since
# it is unclear if that is the actual course or just
# another clarical error, I will drop this course
# too. Given that all we know is that one student
# out of
```

	Course number	Course Name	Course Type
0	ARTS400	EXPERIMENTAL WRITING SEM: The Ecology of Poetry	C
1	ARTS401	ART: ancient to 1945	C
2	ARTS465	ENVIRONMENTAL SYSTEMS II	F
3	ARTS486	COMPUTER LINEAR ALGEBRA	F
4	ARTS512	ANALYTICAL MECHANICS	F
5	ARTS514	A WORLD AT WAR	F
6	ARTS516	BEHAVIORAL PHARMACOLOGY	F
7	ARTS518	CONTEMPORARY AFRICAN ART	F
8	ARTS520	FOOD/FEAST ARCH OF TABLE	F
9	ARTS488	DEVIL'S PACT LIT/FILM	E
10	ARTS541	AMERICAN SOCIAL POLICY	E
11	ARTS543	ART AND RELIGION	E
12	ARTS491	CONTEMPORARY POL.THOUGHT	E
13	ARTS492	AFRICAN-AMERICAN LIT: AFRICAN-AMER LIT:CHANGE	E
14	ARTS493	AMERICAN HEALTH POLICY	E
15	ARTS494	Business German: A Micro Perspective	E
16	ARTS495	COMM and THE PRESIDENCY	E
17	ARTS496	French Thought Till 1945	E
18	ARTS497	CONTEMP ART - 1945 to PRESENT	E
19	ARTS545	20th Century Russian Literature: Fiction and R...	E
20	ARTS547	COMMUNICATIONS INTERNSHP	E
21	ARTS549	FRESHWATER ECOLOGY	E
22	ARTS551	AESTHETICS	E
23	ARTS553	French Thought Since 1945	E
24	ARTS555	BECOMING HUMAN	E
25	ARTS485	EVIDENCED BASED CRIME AND JUSTICE POLICY	E
26	ARTS484	EUROPE IN A WIDER WORLD	E
27	ARTS557	19TH-CENTURY BRITISH LITERATURE	E
28	ARTS559	AMERICAN SOUTH 1861-PRES	E
29	ARTS561	AUGUSTAN CULTRAL REVOLUTION	E
30	ARTS565	Environmental Studies Research Seminar Junior ...	E
31	ARTS567	NaN	E
32	ARTS569	CELL. BIOL. & BIOCHEM.	E
33	ARTS571	FRANCE & THE EUROP.UNION	E
34	ARTS573	ANALYZING THE POL WORLD	E
35	ARTS575	EARLY MESOPOTAM HISTORY/SOCIETY	E
36	ARTS577	FRANCE & THE EUROP.UNION	E
37	ARTS579	EARLY BALCAN HIST/SOC	E
38	ARTS581	COMPARATIVE POLITICS	E
39	ARTS583	BRITISH POETRY 1660-1914	E
40	ARTS585	CONTEMPORARY SOCIO THEORY	E
41	ARTS587	ELEMENTARY ARABIC II	E

	Course number	Course Name	Course Type
31	ARTS567	NaN	E

In [10]:
```
# Remove the nulls since again, we are
# not sure what this class is supposed to be
course_pd_clean = course_pd.dropna(how = 'any')
display(course_pd_clean.iloc[29:33])
```

	Course number	Course Name	Course Type
29	ARTS561	AUGUSTAN CULTRAL REVOLUTION	E
30	ARTS565	Environmental Studies Research Seminar Junior ...	E
32	ARTS569	CELL. BIOL. & BIOCHEM.	E
33	ARTS571	FRANCE & THE EUROP.UNION	E

In [11]:
```
# Let us take a look at how to solve
# the duplicating/repeating class names again
# we can display our unique matches via
# alpha-numerical sorting
sorted(unique_courses)
```

```
 'EYE, MIND AND IMAGE',
 'Environmental Case Studies',
 'Environmental Studies Research Seminar Junior Level',
 'Environmental Studies Research Seminar for Juniors',
 'FICTION WRITING WORKSHOP',
 'FOOD/FEAST ARCH OF TABLE',
 'FORENSIC ANTHROPOLOGY',
 'FORMAL LOGIC I',
 'FORMAL SEM AND COG SCI',
 'FR FOR PROFESSIONS I',
 'FR FOR PROFESSIONS II',
 'FR LIT OF THE 19TH C: STUDIES IN THE 19TH C.',
 'FRANCE & THE EUROP.UNION',
 'FRANCE AND ITS OTHERS: Anthropology and French Modernism',
 'FREEDOM OF EXPRESSION',
 'FRENCH PHONETICS',
 'FRESHWATER ECOLOGY',
 'Feminist Theory: Feminism, Activism, and the Body',
 'French Thought Since 1945']
```

In [12]:
```
# One method to reduce duplicates - this
# is a better case scenario that but there are
# still going to be some missing matches.
# Reducing the threshold starts to cause
# some errorneous matches especially
# with "AMERICAN ..." and "ELEMENTARY ..."

import difflib
import re

for course in unique_courses:
    match = difflib.get_close_matches(course, course_pd_clean['Course Name '], n=1, cutoff=0.85)
    if match:
        print(course)
        print(match[0])

# Since these results look satisfactory
# we can apply the same loop to update
# the whole registration table now -
# we will create a new dataframe with
# the updates to not overwrite the original
reg_courses_cleaned = reg_pd_clean.copy()

i = 0
for course in reg_courses_cleaned['coursename']:
    match = difflib.get_close_matches(course, course_pd_clean['Course Name '], n=1, cutoff=0.85)
    if match:
        reg_courses_cleaned['coursename'].iloc[i] = match[0]
    i += 1
```

```
BEHAVIORAL PHARMACOLOGY
BEHAVIORAL PHARMACOLOGY
COMPUTER LINEAR ALGEBRA
COMPUTER LINEAR ALGEBRA
ANALYTICAL MECHANICS
ANALYTICAL MECHANICS
CONTEMPORARY AFRICAN ART
CONTEMPORARY AFRICAN ART
COMPUT LINEAR ALGEBRA
COMPUTER LINEAR ALGEBRA
A WORLD AT WAR
A WORLD AT WAR
AMERICAN SOUTH 1861-PRES
AMERICAN SOUTH 1861-PRES
ELEMENTARY ARABIC II
ELEMENTARY ARABIC II
AMERICAN HEALTH POLICY
AMERICAN HEALTH POLICY
CONTEMPORARY POL.THOUGHT
```

In [13]:
```
# We can see how the update matched index
# 1766 to the coursename from Course_info.xlsx)
display(reg_pd_clean.iloc[1765:1767])
display(reg_courses_cleaned.iloc[1765:1767])

# We will leave the matching at that for now.
```

	Student name	semester new	coursename
2351	Lorne Michaels	Fall 2001	COMPUT LINEAR ALGEBRA
2352	Lorne Michaels	Fall 2001	FOOD/FEAST ARCH OF TABLE

	Student name	semester new	coursename
2351	Lorne Michaels	Fall 2001	COMPUTER LINEAR ALGEBRA
2352	Lorne Michaels	Fall 2001	FOOD/FEAST ARCH OF TABLE

In [14]:
```
# Question 4 - Which course has the highest registration?
display(reg_courses_cleaned['coursename'].value_counts())

# Computer Linear Algebra appears to be the most common
# As a comparison from our non-matched dataframe
display(reg_pd_clean['coursename'].value_counts())
```

```
coursename
COMPUTER LINEAR ALGEBRA                  325
Environmental Case Studies               286
A WORLD AT WAR                           269
BEHAVIORAL PHARMACOLOGY                  260
ANALYTICAL MECHANICS                     256
                                        ...
ASIAN AMER COMM FLD WRK                    1
FR FOR PROFESSIONS II                      1
ELEM CLASSICAL GREEK II                    1
ANIMAL BEHAVIOR                            1
CREAT.NON-FICTION WRIT: PEER TUTORING      1
Name: count, Length: 151, dtype: int64

coursename
COMPUT LINEAR ALGEBRA                    303
Environmental Case Studies               286
A WORLD AT WAR                           269
BEHAVIORAL PHARMACOLOGY                  260
ANALYTICAL MECHANICS                     256
                                        ...
FR FOR PROFESSIONS II                      1
ELEM CLASSICAL GREEK II                    1
ANIMAL BEHAVIOR                            1
DRUGS, BRAIN, AND MIND                     1
CREAT.NON-FICTION WRIT: PEER TUTORING      1
Name: count, Length: 168, dtype: int64
```

In [15]:
```
# Question 5 - Inner join the two datasets
reg_merged = pd.merge(reg_courses_cleaned,
         course_pd_clean.rename(columns={'Course Name ': 'coursename'}),
         on='coursename', how='inner')

# Using merge, we can now see the full frame
# of each student's registration containing
# not only the course name but also number and
# type

display(reg_merged)
```

	Student name	semester new	coursename	Course number	Course Type
0	Bill Mumy	Fall 2004	BEHAVIORAL PHARMACOLOGY	ARTS516	F
1	Geraldine Ferraro	Summer 2004	BEHAVIORAL PHARMACOLOGY	ARTS516	F
2	Laura Lippman	Fall 2004	BEHAVIORAL PHARMACOLOGY	ARTS516	F
3	Dom DeLuise	Fall 2000	BEHAVIORAL PHARMACOLOGY	ARTS516	F
4	Sally Field	Summer 2001	BEHAVIORAL PHARMACOLOGY	ARTS516	F
...	...	...	...	...	...
2233	Pamela Jones	Fall 2001	CONTEMP ART - 1945 to PRESENT	ARTS497	E
2234	Rita Moreno	Fall 2001	CONTEMP ART - 1945 to PRESENT	ARTS497	E
2235	Tony Blair	Fall 2004	CONTEMP ART - 1945 to PRESENT	ARTS497	E
2236	Edward Koch	Fall 2004	CONTEMP ART - 1945 to PRESENT	ARTS497	E
2237	Betty Hutton	Fall 2000	EARLY MESOPOTAM HISTORY/SOCIETY	ARTS575	E

2238 rows × 5 columns

In [39]:
```
# Question 6 - Create a dataframe containing course numbers as columns
# First, we will create a dummy table containing the respective desired
# hiearchy
data = pd.DataFrame(index=pd.Index(sorted(unique_students), name='Student'),
                    columns=pd.Index(sorted(reg_merged['Course number'].unique()), name='Course'))

# For iterating, we will copy the merged table student names
unique_st_clean = reg_merged['Student name']

# We will double loop over each student and find what courses they are registered to
for st in sorted(unique_st_clean):
# Not the cleanest method but we will create a temp frame for each student
    reg_tmp = reg_merged[reg_merged['Student name'].str.match(st)]
# Now loop over each course number returned to the respective student
    for cs in reg_tmp['Course number'].values:
# Set the location in the original table to true now
        data[cs].loc[st] = 1

# Change all nulls to 0
data = data.fillna(0)
```

In [40]:
```
data
```

Out[40]:

Course	ARTS400	ARTS401	ARTS465	ARTS484	ARTS485	ARTS486	ARTS488	ARTS491	ARTS492	ARTS493	...	ARTS565	ARTS569	ARTS571	ARTS573	ARTS575	ARTS577	ARTS581	ARTS583	ARTS585
Student																				
ABella Abzug	0	1	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	1	0	0
Al Gore	0	0	0	0	0	1	0	0	0	1	...	0	0	0	0	0	0	0	0	0
Al Hirt	0	0	1	0	0	1	0	0	0	0	...	0	1	0	0	0	0	0	0	0
Al Roker	1	0	0	0	0	1	0	0	0	1	...	0	0	0	0	0	0	0	0	0
Alan Alda	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Winona Ryder	0	0	0	0	0	1	0	0	0	0	...	0	1	0	0	0	0	0	0	0
Wolfgang Puck	0	0	0	0	0	1	0	0	0	1	...	0	0	0	0	0	0	0	0	0
Yogi Berra	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
Yoko Ono	0	0	0	0	0	1	0	0	0	0	...	0	0	0	0	0	0	0	0	0
state representative	0	1	0	0	0	1	0	0	0	1	...	0	0	0	0	0	0	0	0	0

448 rows × 37 columns

In [ ]:
```

```