In [1]:
```
#import libraries
import pandas as pd

# ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
# load datasets
a = pd.read_csv('Answers.csv', usecols = ['Id', 'OwnerUserId', 'CreationDate', 'ParentId', 'Scor
#print(a.shape)
#print(a.info())
#a.head()
q = pd.read_csv('Questions.csv', usecols = ['Id', 'OwnerUserId', 'CreationDate', 'ClosedDate', '
#print(q.shape)
#print(q.info())
#q.head()
t = pd.read_csv('Tags.csv', usecols = ['Id', 'Tag'], low_memory=False)
#print(t.shape)
#print(t.info())
#t.head()
```

Clean the Data

We now have the data from Kaggle loaded into three dataframes. We will first clean up the data by removing any rows with invalid numerical values and converting then into integer types. We will also remove any columns that we don't need.

In [3]:
```
#clean questions dataframe
q_clean = q
# drop any rows where ID contains non-numerical values
q_clean = q_clean[q_clean['Id'].str.contains('\D') == False]
q_clean['Id'] = q_clean['Id'].astype(str).astype(int)
# drop any rows where Score contains non-numerical values
q_clean = q_clean[q_clean['Score'].str.contains('\D') == False]
q_clean['Score'] = q_clean['Score'].astype(str).astype(int)
#remove unwanted columns
q_clean.drop(["OwnerUserId"], axis=1, inplace=True)
print(q_clean.info())

#clean answers dataframe
a_clean = a
#remove unwanted columns
a_clean.drop(["Id", "OwnerUserId"], axis=1, inplace=True)
a_clean.rename(columns={'ParentId' : 'Id'}, inplace=True)
print(a_clean.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 976367 entries, 0 to 1048574
Data columns (total 6 columns):
 #   Column        Non-Null Count   Dtype 
---  ------        --------------   ----- 
 0   Id            976367 non-null  int32 
 1   CreationDate  976367 non-null  object
 2   ClosedDate    31217 non-null   object
 3   Score         976367 non-null  int32 
 4   Title         976367 non-null  object
 5   Body          976366 non-null  object
dtypes: int32(2), object(4)
memory usage: 44.7+ MB
None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 4 columns):
 #   Column        Non-Null Count    Dtype 
---  ------        --------------    ----- 
 0   CreationDate  1048575 non-null  object
 1   Id            1048575 non-null  int64 
 2   Score         1048575 non-null  int64 
 3   Body          1048575 non-null  object
dtypes: int64(2), object(2)
memory usage: 32.0+ MB
None
```

Reduce to Relevant Data

There is much more data here than we need, as we are focusing on posts regarding the following programming languages: java, c++, and python. We must remove the data we won't be using.
In order to do this, we will first create three new dataframes from the tag dataset with the tags java, python, and c++.

In [4]:
```python
#extract relevent posts by tags
# t.info()
# t['Tag'].value_counts(ascending=False).reset_index().head(15)
t_java = pd.DataFrame(t[t.Tag == 'java'])
print("Java: ", t_java.shape)
print(t_java.head())

t_python = pd.DataFrame(t[t.Tag == 'python'])
print("Python: ", t_python.shape)
print(t_python.head())

t_cpp = pd.DataFrame(t[t.Tag == 'c++'])
print("C++: ", t_cpp.shape)
print(t_cpp.head())
```

```
Java:  (29266, 2)
        Id   Tag
127   4080  java
145   4630  java
211   7720  java
304  10980  java
346  11930  java
Python:  (13470, 2)
         Id     Tag
312   11060  python
503   17250  python
546   19030  python
905   31340  python
1027  34020  python
C++:  (14691, 2)
        Id  Tag
18     330  c++
107   3150  c++
112   3230  c++
216   7880  c++
302  10880  c++
```

Reduce to Relevant Data pt 2

We will then use the IDs in the tag dataframes to extract the corresponding rows from the questions and answers dataframes. We will end up with six dataframes: two (questions and answers) for each language (java, c++, python)

In [5]:
```
questions_java = pd.merge(t_java, q_clean, how='inner', on=['Id'])
#questions_java.head()
answers_java = pd.merge(t_java, a_clean, how='inner', on=['Id'])
#answers_java.head()

questions_python = pd.merge(t_python, q_clean, how='inner', on=['Id'])
#questions_python.head()
answers_python = pd.merge(t_python, a_clean, how='inner', on=['Id'])
#answers_python.head()

questions_cpp = pd.merge(t_cpp, q_clean, how='inner', on=['Id'])
#questions_cpp.head()
answers_cpp = pd.merge(t_cpp, a_clean, how='inner', on=['Id'])
#answers_cpp.head()

# to csv:
questions_java.to_csv("questions_java.csv")
questions_python.to_csv("questions_python.csv")
questions_cpp.to_csv("questions_cpp.csv")
answers_java.to_csv("answers_java.csv")
answers_python.to_csv("answers_python.csv")
answers_cpp.to_csv("answers_cpp.csv")
```

In [ ]:
```
```