**Step 3: In groups, create a Jupyter notebook that shows the mortgage problem solved in Python. Make sure to:**

- Illustrate how your data structure is used to solve the problem from GWP 1.
- Ensure that your results match the results from the spreadsheet.

In [1]:
```
import pandas as pd
import numpy as np
import datetime

pd.options.display.float_format = '${:,.2f}'.format
```

## Cashflow is calcualted by formular:

$$\text{Cashflow} = \frac{\text{Principal} \times \text{Ratemonthly}}{1 - \frac{1}{(1+\text{Ratemonthly})^{\text{Year} \times 12}}}$$

In [2]:
```
def mortgage(year, rate=0, start_date=None, principal=1000000):

    period = 12*year
    remainingprincipal = 0

    if start_date is None:
        pass
    else:
        start_date = pd.to_datetime(start_date)
        # initial array of period in monthly from start date
        rangedatemonth = pd.date_range(
            start=start_date-pd.DateOffset(months=1), periods=period+1, freq='MS')

        ratedf = pd.read_csv("https://docs.google.com/spreadsheets/d/e/"
                             "2PACX-1vS5svAeBupJJ5CsK94DPLLTTVB4ZuckMMi-suaBzzajdfbe2"
                             "amCffYEz9KysCItneZ7Jk65eesKFWTN/pub?output=csv"
                             )
        ratedf = ratedf.iloc[:, 1:3]
        ratedf.DATE = pd.to_datetime(ratedf.DATE)
        ratedf.sort_values("DATE", inplace=True)
        ratedf['Year'] = pd.DatetimeIndex(ratedf.DATE).year

        raterange = []
        for d in rangedatemonth:
            if d.year < (start_date.year+7):
                raterange.append(
                    ratedf[~(ratedf['Year'] >= start_date.year)][-1:].MORTGAGE30US.to_list()[0]/100)
            else:
                # year duration - 7 years
                raterange.append(
                    ratedf[~(ratedf['Year'] >= d.year)][-1:].MORTGAGE30US.to_list()[0]/100)

    frame = []

    for p in range(period+1):
        openningprincipal = remainingprincipal
        if start_date != None:
            rate = raterange[p]

        if p == 0:
            cashflow = 0
            interestpaid = 0
            principalpaid = cashflow-interestpaid
            remainingprincipal = principal
            frame.append([p, principal, rate, cashflow,
                          interestpaid, principalpaid, remainingprincipal])
        else:
            if (p >= 12*7) and (p % 12 == 1):
                principal = remainingprincipal
                period = year*12 - p + 1
            if period == 0:
                cashflow = openningprincipal
            else:
                cashflow = (principal * rate/12) / \
                    (1 - 1/((1+rate/12)**(period)))
            interestpaid = openningprincipal*rate/12
            principalpaid = cashflow-interestpaid
            remainingprincipal = openningprincipal-principalpaid
            frame.append([p, openningprincipal, rate, cashflow,
                          interestpaid, principalpaid, remainingprincipal])

    df = pd.DataFrame(frame, columns=["Month", "Opening Principal Balance", "Rates", "Cashflow",
                                      "Interest Paid", "Principal Paid", "Closing Princial Balance"])
    df.set_index("Month", inplace=True)

    df.Rates = (df.Rates*100).map('{:,.2f}%'.format)

    return df
```

## Mortgage 1:

- Time: 30 years
- Rate: 4%
- Principal Amount: $1,000,000

In [3]:
```
Mortgage1 = mortgage(30,0.04)
print("Total Interest Paid: ${:#,.2f}".format(Mortgage1['Interest Paid'].sum()))
Mortgage1
```

```
Total Interest Paid: $718,695.06
```

Out[3]:

| Month | Opening Principal Balance | Rates | Cashflow | Interest Paid | Principal Paid | Closing Princial Balance |
|---|---|---|---|---|---|---|
| **0** | $1,000,000.00 | 4.00% | $0.00 | $0.00 | $0.00 | $1,000,000.00 |
| **1** | $1,000,000.00 | 4.00% | $4,774.15 | $3,333.33 | $1,440.82 | $998,559.18 |
| **2** | $998,559.18 | 4.00% | $4,774.15 | $3,328.53 | $1,445.62 | $997,113.56 |
| **3** | $997,113.56 | 4.00% | $4,774.15 | $3,323.71 | $1,450.44 | $995,663.12 |
| **4** | $995,663.12 | 4.00% | $4,774.15 | $3,318.88 | $1,455.28 | $994,207.84 |
| **...** | ... | ... | ... | ... | ... | ... |
| **356** | $23,633.90 | 4.00% | $4,774.15 | $78.78 | $4,695.37 | $18,938.53 |
| **357** | $18,938.53 | 4.00% | $4,774.15 | $63.13 | $4,711.02 | $14,227.50 |
| **358** | $14,227.50 | 4.00% | $4,774.15 | $47.43 | $4,726.73 | $9,500.78 |
| **359** | $9,500.78 | 4.00% | $4,774.15 | $31.67 | $4,742.48 | $4,758.29 |
| **360** | $4,758.29 | 4.00% | $4,774.15 | $15.86 | $4,758.29 | $0.00 |

361 rows × 6 columns

## Mortgage 2:

- Time: 20 years
- Rate: 2.5%
- Principal Amount: $1,000,000

In [4]:
```
Mortgage2 = mortgage(20,0.025)
print("Total Interest Paid: ${:#,.2f}".format(Mortgage2['Interest Paid'].sum()))
Mortgage2
```

```
Total Interest Paid: $271,766.94
```

Out[4]:

| Month | Opening Principal Balance | Rates | Cashflow | Interest Paid | Principal Paid | Closing Princial Balance |
|---|---|---|---|---|---|---|
| **0** | $1,000,000.00 | 2.50% | $0.00 | $0.00 | $0.00 | $1,000,000.00 |
| **1** | $1,000,000.00 | 2.50% | $5,299.03 | $2,083.33 | $3,215.70 | $996,784.30 |
| **2** | $996,784.30 | 2.50% | $5,299.03 | $2,076.63 | $3,222.39 | $993,561.91 |
| **3** | $993,561.91 | 2.50% | $5,299.03 | $2,069.92 | $3,229.11 | $990,332.80 |
| **4** | $990,332.80 | 2.50% | $5,299.03 | $2,063.19 | $3,235.84 | $987,096.97 |
| **...** | ... | ... | ... | ... | ... | ... |
| **236** | $26,330.35 | 2.50% | $5,299.03 | $54.85 | $5,244.17 | $21,086.18 |
| **237** | $21,086.18 | 2.50% | $5,299.03 | $43.93 | $5,255.10 | $15,831.08 |
| **238** | $15,831.08 | 2.50% | $5,299.03 | $32.98 | $5,266.05 | $10,565.03 |
| **239** | $10,565.03 | 2.50% | $5,299.03 | $22.01 | $5,277.02 | $5,288.01 |
| **240** | $5,288.01 | 2.50% | $5,299.03 | $11.02 | $5,288.01 | $0.00 |

241 rows × 6 columns

## Mortgage 3:

- Time: 30 years
- Rate: 7-1 Adjustable
- Principal Amount: $1,000,000

In [5]:
```
Mortgage3 = mortgage(year=30,start_date='1990-01-01')
print("Total Interest Paid: ${:#,.2f}".format(Mortgage3['Interest Paid'].sum()))
Mortgage3
```

```
Total Interest Paid: $1,488,119.85
```

Out[5]:

| Month | Opening Principal Balance | Rates | Cashflow | Interest Paid | Principal Paid | Closing Princial Balance |
|---|---|---|---|---|---|---|
| **0** | $1,000,000.00 | 9.78% | $0.00 | $0.00 | $0.00 | $1,000,000.00 |
| **1** | $1,000,000.00 | 9.78% | $8,613.58 | $8,150.00 | $463.58 | $999,536.42 |
| **2** | $999,536.42 | 9.78% | $8,613.58 | $8,146.22 | $467.36 | $999,069.05 |
| **3** | $999,069.05 | 9.78% | $8,613.58 | $8,142.41 | $471.17 | $998,597.88 |
| **4** | $998,597.88 | 9.78% | $8,613.58 | $8,138.57 | $475.01 | $998,122.87 |
| **...** | ... | ... | ... | ... | ... | ... |
| **356** | $29,374.18 | 4.55% | $5,941.83 | $111.38 | $5,830.45 | $23,543.73 |
| **357** | $23,543.73 | 4.55% | $5,941.83 | $89.27 | $5,852.56 | $17,691.17 |
| **358** | $17,691.17 | 4.55% | $5,941.83 | $67.08 | $5,874.75 | $11,816.42 |
| **359** | $11,816.42 | 4.55% | $5,941.83 | $44.80 | $5,897.03 | $5,919.39 |
| **360** | $5,919.39 | 4.55% | $5,941.83 | $22.44 | $5,919.39 | $-0.00 |

361 rows × 6 columns