

# Financial services information system

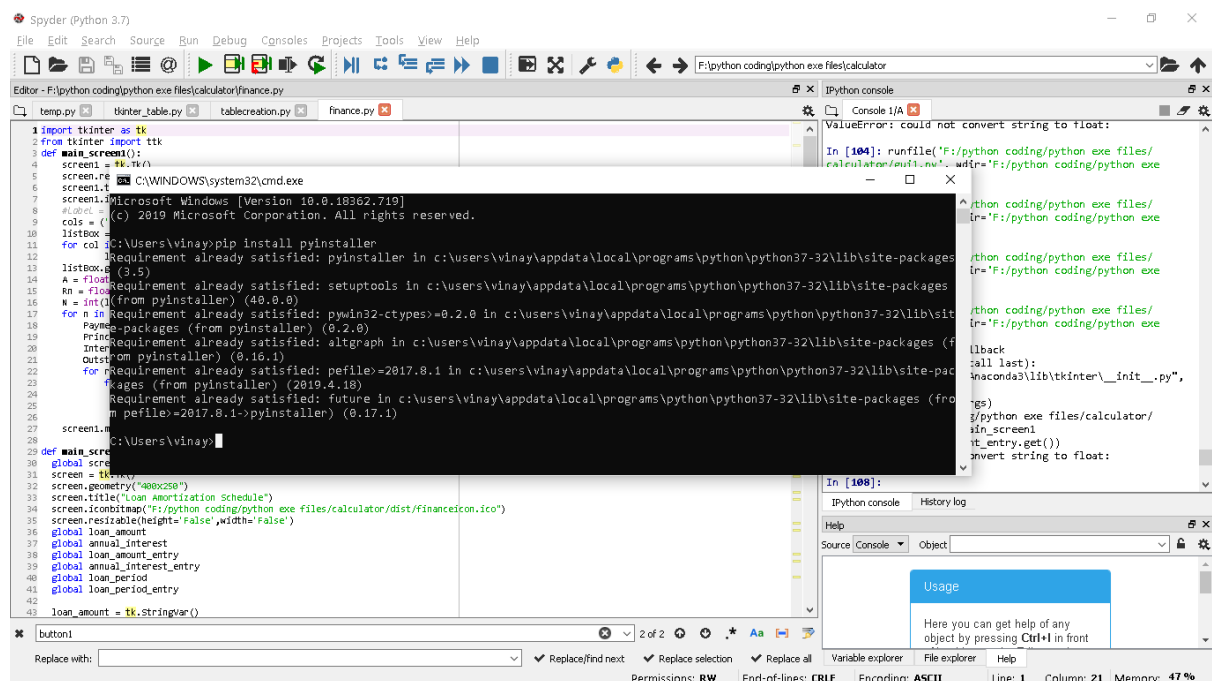
## Assignment-2

### 1. Creating GUI (graphical user interface) for Loan amortization schedule

The entire GUI is build base on python. To generate new GUI from finance.py file import the following packages.

```
import tkinter as tk
from tkinter import ttk
```

to create an executable file import pyinstaller using command 'pip install pyinstaller' and run these command in command prompt



The screenshot shows the Spyder Python IDE interface. The editor window displays the code for 'finance.py', which includes imports for tkinter and ttk, and a main function that sets up a Tkinter window titled 'Loan Amortization Schedule'. The IPython console shows the output of running the script, which includes the command 'pip install pyinstaller' and the execution of 'runfile' for the 'calculator' script. A 'Usage' dialog box is also visible in the bottom right corner.

Similarly, for tkinter use “pip install tkinter” and use the command “pyinstaller –F finance.py” for converting finanac.py to finance.exe

After complete the setup for the python

### 2. Logic for Loan amortization schedule

Calculate the payments for Loan amortization schedule.

Total amount of loan (A):

Annual interest rate (Rn):

Number of loan periods (N): (in years or months)

Formulas:

```

Payment_amount = (A*Rn*((1+Rn)**N))/(((1+Rn)**N)-1)

Principal_amount = (Payment_amount*(((1+Rn)**(n-N-1))))

Interest_amount = (Payment_amount - Principal_amount)

Outstanding_amount = ((Interest_amount/Rn) - Principal_amount)

```

### Function used in finance.py is Loan amortization schedule

```

def amortization_schedule():
    A = float(input("Enter the total loan amount :"))
    Rn = float(input("Enter annual payment :")) / (100*12)
    N = int(input("Enter the total period(in years) :"))*12
    for n in range(1, (N+1), 1):
        Payment_amount = (A*Rn*((1+Rn)**N))/(((1+Rn)**N)-1)
        Principal_amount = (Payment_amount*(((1+Rn)**(n-N-1))))
        Interest_amount = (Payment_amount - Principal_amount)
        Outstanding_amount = ((Interest_amount/Rn) - Principal_amount)

```

### 3. GUI design for Loan amortization schedule

The user interface for entering values for Loan amortization

```

tk.Label(screen, text = "Please enter details below",font=(
    "times new roman",15,"bold")).pack()
tk.Label(screen, text = "Total amount of loan :",font=("verdana italic",10,"bold")).pack()
loan_amount_entry = tk.Entry(screen, textvariable =
loan_amount,bd=5,relief="raised",bg="white",fg="black")
loan_amount_entry.pack()

tk.Label(screen, text = "Annual interest rate :",font=("verdana italic",10,"bold")).pack()
annual_interest_entry = tk.Entry(screen, textvariable =
annual_interest,bd=5,relief="raised",bg="white",fg="black")
annual_interest_entry.pack()

tk.Label(screen, text = " Number of loan period(years):",font=("verdana
italic",10,"bold")).pack()
loan_period_entry = tk.Entry(screen, textvariable =
loan_period,bd=5,relief="raised",bg="white",fg="black")
loan_period_entry.pack()
tk.Label(screen, text = "").pack()
ttk.Button(text = "submit", width = "15", command = amortization_schedule).pack()
ttk.Style().configure("TButton", padding=6, relief="flat", background="#000",font=("verdana
italic",10,"bold"))

```

- tk.Label() are used to create the fields and field names
- ttk.Button() are used to create the buttons

- tk.Label() and ttk.Button() has different attributes such as text, relief , background ,font etc

**Please enter details below**

*Total amount of loan :*

*Annual interest rate :*

*Number of loan period(years):*

**submit**

The result table for loan amortization

```
cols = ('Payment No', 'Payment Amount', 'Principal amount paid', 'Interest amount paid', 'Loan outstanding balance')
listBox = ttk.Treeview(screen1, columns=cols, show='headings')
```

- ttk.Treeview is function used represent the data in particular format

Payment No	Payment Amount	Principal amount paid	Interest amount paid	Loan outstanding balance
1	5368.216230121	1201.549563455	4166.666666667	998798.45043654
2	5368.216230121	1206.556019969	4161.660210152	997591.89441657
3	5368.216230121	1211.583336719	4156.632893402	996380.31107985
4	5368.216230121	1216.631600622	4151.584629499	995163.67947923
5	5368.216230121	1221.700898958	4146.515331163	993941.97858023
6	5368.216230121	1226.79131937	4141.424910751	992715.18726090
7	5368.216230121	1231.902949868	4136.313280254	991483.28431103
8	5368.216230121	1237.035978825	4131.180351296	990246.24843221
9	5368.216230121	1242.190194987	4126.026035134	989004.05823722
10	5368.216230121	1247.365987466	4120.850242655	987756.69224976

### Integration logic for interface design and amortization\_schedule function

- The values entered in the fields total amount, annual interest rate and number of years are integrated with table values

### Values integration

```
A = float(loan_amount_entry.get())
Rn = float(annual_interest_entry.get())/(100*12)
N = int(loan_period_entry.get())*12
```

### Functional logic integration😊

```
for row in range(n,n+1):
    for column in range(1):
        tempList = [[row], [round(Payment_amount,9)], [round(Principal_amount,9)],
                    [round(Interest_amount,9)], [abs(round(Outstanding_amount,9))]]
        listBox.insert("", "end", values=(tempList))
```