Instructor Name: Rahul Tiwari

Email: support@rahultiwari.co.in

```
0-2500 -> 0%
2501 - 5000 -> 5%
5001 - 10000 -> 10%
10001 - 20000 -> 20%
20001+ -> 30%
```

Tax Calculations

```
salary = int(input('Enter the salary: '))
if salary<=2500:
    print('The total tax you have to pay:', salary*0.00)
elif salary>2500 and salary<=5000:
    print('The total tax you have to pay:', salary*0.05)
elif salary>5000 and salary<=10000:
    print('The total tax you have to pay:', salary*0.10)
elif salary>10000 and salary<=20000:
    print('The total tax you have to pay:', salary*0.20)
else:
    print('The total tax you have to pay:', salary*0.30)
```

```
Enter the salary: 6000
The total tax you have to pay: 600.0
```

```
def tax_calc(salary):
    if salary<=2500:
        return('The total tax you have to pay:', salary*0.00)
    elif salary>2500 and salary<=5000:
        return('The total tax you have to pay:', salary*0.05)
    elif salary>5000 and salary<=10000:
        return('The total tax you have to pay:', salary*0.10)
    elif salary>10000 and salary<=20000:
        return('The total tax you have to pay:', salary*0.20)
    else:
        return('The total tax you have to pay:', salary*0.30)

output = tax_calc(1000)

output
```

```
('The total tax you have to pay:', 0.0)
```

Account Balance Check

```
account_balance = int(input('Enter the account balance: '))
if account_balance>=0:
    print('Your account is in good standing')
```

```
else:
    print('Your account balance is negative, please take action')
```

```
Enter the account balance: -10
Your account balance is negative, please take action
```

Loan Eligibility

```
income = int(input('Enter the income: '))
credit_score = int(input('Enter the credit score: '))

if income>50000 and credit_score>700:
    print('You are eligible for the loan')
else:
    print('You are not eligible for the loan')
```

```
Enter the income: 60000
Enter the credit score: 800
You are eligible for the loan
```

Transaction Fees Calculation - Nested If and else

```
transaction_amount = int(input('Enter the transaction amount: '))
transaction_type = input('Enter the transaction type: ')
if transaction_type=='Withdraw':
    if transaction_amount>1000:
        transaction_fee = 5
    elif transaction_amount<=1000:
        transaction_fee = 2
elif transaction_type=='deposit':
    transaction_fee = 0
else:
    print('Please pass the correct option')

total_amount = transaction_amount+transaction_fee
print('Total Amount: ', total_amount)
```

```
Enter the transaction amount: 500
Enter the transaction type: Withdraw
Total Amount:  502
```

```
company = 'Upgrad'
for i in range(0,len(company)):
    print(company[i])
```

```
U
p
g
r
```

```
a
d

list(range(0,len(company)))

[0, 1, 2, 3, 4, 5]

sales = [3211,2311,1232,1221]
profit = [324,223,221,213]
for i in range(0,len(sales)):
    print(profit[i]/sales[i])

0.10090314543755839
0.09649502379922112
0.1793831168831169
0.17444717444717445
```

Transaction History

```
transaction_history = [50,-20,100,-30,-10]
total_balance = 0
print('Transaction History')
for i in transaction_history:
    if i>0:
        print('Deposit:', i)
    else:
        print('Withdraw: ', abs(i))
    total_balance = total_balance+i
print('Total balance: ',total_balance)

Transaction History
Deposit: 50
Withdraw:  20
Deposit: 100
Withdraw:  30
Withdraw:  10
Total balance:  90
```

Interest Calculations

```
account_balance = [1000,2000,3000,4000,2110,3000]
interest_rate = 0.05

for i in range(len(account_balance)):
    interest = account_balance[i] * interest_rate
    #account_balance[i]+=interest
    account_balance[i]=account_balance[i]+interest
print('Updated Balance: ', account_balance)

Updated Balance:  [1050.0, 2100.0, 3150.0, 4200.0, 2215.5, 3150.0]
```

Account Closure

```
account_numbers = [123456,789012,345678,901234]
account_to_close = int(input('Enter the account number to close: '))

for account in account_numbers:
    if account==account_to_close:
        print('Closing account: ', account)
        account_numbers.remove(account)
        break
    else:
        print('Account not found')

print('Remaining accounts: ', account_numbers)
```

```
Enter the account number to close: 222222
Account not found
Account not found
Account not found
Account not found
Remaining accounts:  [123456, 789012, 345678, 901234]
```

```
account_numbers = [123456,789012,345678,901234]
account_to_close = int(input('Enter the account number to close: '))

for account in account_numbers:
    if account==account_to_close:
        print('Closing account: ', account)
        account_numbers.remove(account)
        break

print('Remaining accounts: ', account_numbers)
```

```
Enter the account number to close: 123456
Closing account:  123456
Remaining accounts:  [789012, 345678, 901234]
```

```
account_numbers = [123456,789012,345678,901234]
account_to_close = int(input('Enter the account number to close: '))

new_list = account_numbers.copy()
for account in account_numbers:
    if account==account_to_close:
        print('Closing account: ', account)
        account_numbers.remove(account)
        break

print('Remaining accounts: ', account_numbers)
if new_list==account_numbers:
    print('We did not find the account')
```

```
else:
    print('Account removed successfully', account_to_close)
```

```
Enter the account number to close: 222222
Remaining accounts:  [123456, 789012, 345678, 901234]
We did not find the account
```

Account Statement

```
account_transactions = [{"date":"2023-07-
01","type":"Deposit","amount":100},
                        {"date":"2023-07-
02","type":"Withdraw","amount":50},
                        {"date":"2023-07-
03","type":"Deposit","amount":200},
                        {"date":"2023-07-
04","type":"Withdraw","amount":75}]

print("Account Statement")
for transaction in account_transactions:
    print(transaction['date'], '-', transaction['type'], '-',
transaction['amount'])
```

```
Account Statement
2023-07-01 - Deposit - 100
2023-07-02 - Withdraw - 50
2023-07-03 - Deposit - 200
2023-07-04 - Withdraw - 75
```

```
transaction_date = ["2023-07-01","2023-07-02","2023-07-03","2023-07-
04"]
transaction_type = ["Deposit","Withdraw","Deposit","Withdraw"]
transaction_amount = [100,50,200,75]
print('Account Statement')
for (td,tt,ta) in
zip(transaction_date,transaction_type,transaction_amount):
    print(td,'-',tt,'-',ta)
```

```
Account Statement
2023-07-01 - Deposit - 100
2023-07-02 - Withdraw - 50
2023-07-03 - Deposit - 200
2023-07-04 - Withdraw - 75
```

While Loops

Python Coupoun code

```
coupoun = 5
utilized = 0
```

```
while utilized<coupoun:
    print('The total coupouns left are: ', coupoun-utilized)
    utilized=utilized+1
print('No coupouns left')
```

```
The total coupouns left are:  5
The total coupouns left are:  4
The total coupouns left are:  3
The total coupouns left are:  2
The total coupouns left are:  1
No coupouns left
```

Factorial of a number

```
number = int(input('Enter the number: '))
if number<0:
    print('There is no factorial of the negative number')
else:
    result = 1
    while number>0:
        result=result*number
        number=number-1
    print(result)
```

```
Enter the number: 0
1
```

Pin validation

```
correct_pin = 1234
max_attempts = 3
attempts = 0

while attempts<max_attempts:
    pin = int(input('Enter the pin: '))
    if pin==correct_pin:
        print("Pin accepted, access granted")
        break
    else:
        attempts+=1
        print("Incorrect Pin attempts remaining: ", max_attempts-
attempts)

if max_attempts==attempts:
    print('Max Attempts reached, access denied')
```

```
Enter the pin: 1234
Pin accepted, access granted
```

```
correct_pin = 1234
max_attempts = 3
attempts = 0
account_balance = 0
while attempts<max_attempts:
    pin = int(input('Enter the pin: '))
    if pin==correct_pin:
        print("Pin accepted, access granted")
        while True:
            transaction_type = input('Enter the transaction type: ')
            if transaction_type=='Deposit':
                amount = int(input('Enter the amount to deposit: '))
                account_balance+=amount
                print('Your account balance:', account_balance)
            elif transaction_type=='Withdraw':
                amount = int(input('Enter the amount to withdraw: '))
                if amount>account_balance:
                    print('Insufficient funds')
                    break
                else:
                    account_balance-=amount
    else:
        attempts+=1
        print("Incorrect Pin attempts remaining: ", max_attempts-
attempts)

if max_attempts==attempts:
    print('Max Attempts reached, access denied')
```

```
Enter the pin: 1234
Pin accepted, access granted
Enter the transaction type: Deposit
Enter the amount to deposit: 30000
Your account balance: 30000
Enter the transaction type: Withdraw
Enter the amount to withdraw: 5000
Enter the transaction type: Deposit
Enter the amount to deposit: 2000
Your account balance: 27000
Enter the transaction type: Withdraw
Enter the amount to withdraw: 28000
Insufficient funds
Enter the pin: 2222
Incorrect Pin attempts remaining:  2
Enter the pin: 222
Incorrect Pin attempts remaining:  1
Enter the pin: 222
Incorrect Pin attempts remaining:  0
Max Attempts reached, access denied
```

```
account_balance

1000

account

901234
```

Create Unique Account Number

```
import random
existing_accounts = [1,3,7,9]
new_account_number = random.randint(1,9)

while new_account_number in existing_accounts:
    new_account_number = random.randint(1,9)

print("Account Created Successfully: ", new_account_number)

Account Created Successfully:  8
```

Interest Calculations

```
Account_Balance = 200000
interest_rate = 0.05
max_duration = 104
current_week = 1
while current_week<=max_duration:
    interest = Account_Balance*interest_rate
    Account_Balance+=interest
    current_week+=1

print("Final corpus we will have after 104 weeks:", Account_Balance)

Final corpus we will have after 104 weeks: 31968120.15900867

Account_Balance = 200000
interest_rate = 0.6
max_duration = 2
current_week = 1
while current_week<=max_duration:
    interest = Account_Balance*interest_rate
    Account_Balance+=interest
    current_week+=1

print("Final corpus we will have after 24 months:", Account_Balance)

Final corpus we will have after 24 months: 512000.0
```

Captcha Generation

```python
import random
def generate_captcha():
    def generate_random_digit():
        return random.randint(0,9)

    def generate_random_letter():
        return chr(random.randint(97,122))

    captcha = ""
    for i in range(6):
        if random.choice([True,False]) == True:
            captcha += str(generate_random_digit())
        else:
            captcha +=generate_random_letter()

    print('Generated Captcha: ', captcha)

generate_captcha()
```

```
Generated Captcha:  2tmf11
```