

Experiment No. 2: Tax Calculator

2.1 Objectives

2.1.1 General Objective

To demonstrate the use of python to solve for the monthly and annual tax of one individual.

2.2 Theory

2.2.1 Tax Table

TRAIN, or Tax Reform for Acceleration and Inclusion, is a new tax reform law that was signed into law during the Duterte administration in the Philippines on December 19, 2017, and it went into effect on January 1, 2018. From year 2023 onwards, the income tax rates will be shown as follows:

The image shows a graphic titled "Income Tax Tables under TRAIN Law (Part 2 – Applicable from Year 2023 Onwards)". It contains a table with two columns: "ANNUAL INCOME" and "TAX RATE". The table lists six income brackets and their corresponding tax rates. The source is cited as "Source: BIR and www.PinoyMoneyTalk.com". At the bottom, there is a logo for PinoyMoneyTalk.com.

ANNUAL INCOME	TAX RATE
P250,000 and below	None (0%)
Above P250,000 to P400,000	15% of excess over P250,000
Above P400,000 to P800,000	P22,500 + 20% of excess over P400,000
Above P800,000 to P2,000,000	P102,500 + 25% of excess over P800,000
Above P2,000,000 to P8,000,000	P402,500 + 30% of excess over P2,000,000
Above P8,000,000	P2.2025 million + 35% of excess over P8 million

Figure 2. 1 Tax Table

2.3 Syntax and Functions

2.3.1 Input Function

Usage: Allows user to input variables

Syntax:

```
input(prompt)
```

2.3.2 Print Function

Usage: Display words/sentences for user to read

Syntax:

```
print("example")
```

2.3.3 Functions

Usage: Blocks of code designed to do one specific job that can be used again and again, rather than typing it multiple times

Syntax:

```
def "variable":
```

2.4 Methodology

```
import os
```

```
monthlyIncome = float(input("Enter monthly income: "))  
annualIncome = monthlyIncome*12
```

```
def taxCalc(annualIncome):  
    if annualIncome <= 250000:  
        return annualIncome*0 #0% Tax Rate  
    elif annualIncome <= 400000:  
        return annualIncome*.15 #15% Tax Rate  
    elif annualIncome <= 800000:  
        return annualIncome*.20+22500 #20% Tax Rate  
    elif annualIncome <= 2000000:  
        return annualIncome*.25+102500 #25% Tax Rate  
    elif annualIncome <= 8000000:  
        return annualIncome*.30+402500 #30% Tax Rate  
    else:  
        return annualIncome*.35+2205000
```

```
monthlyTax = taxCalc(annualIncome)/12  
annualTax = taxCalc(annualIncome)
```

```
monthlyNetPay = monthlyIncome-monthlyTax  
annualNetPay = annualIncome-annualTax
```

```
print ("\nYour monthly income is: {}".format (monthlyIncome))  
print ("Your annual income is: {}".format(annualIncome))
```

```
print ("\nYour monthly tax is: {}".format (monthlyTax))  
print ("Your annual tax is: {}".format(annualTax))
```

```

print ("\nYour monthly net pay: {}".format(monthlyNetPay))
print ("Your annual net pay is: {}".format(annualNetPay))

print("\nThank you for using this program!\n")

os.system("pause")

```

2.5 Results and Discussion

Table 1.1 Summary of results for computing for Monthly and Annual Tax

Monthly Income	Annual Income	Manual Computation (Monthly Tax)	Manual Computation (Annual Tax)	Python Code (Monthly Tax)	Python Code (Annual Tax)
50000	600000	11875	142500	11875.0	142500.0
25000	300000	3750	45000	3750.0	45000.0

From the results above, it can be concluded that the program created is working correctly. This is said because all the manual computations are the same as the ones computed from the program created with python.