

**Compilation of Laboratory Experiments**

Presented to the

Department of Manufacturing Engineering and Management

Gokongwei College of Engineering

De La Salle University

In Partial Fulfillment of the Requirements

for Programming Logic and Design Laboratory

Submitted by:

**De La Rosa, Eliza Rica**BS-CPE| 12239577

Instructor:

**Engr. Immanuel Jose C. Valencia**

2023

Contents

[List of Figures iv](#_Toc129643807)

[List of Tables v](#_Toc129643808)

[1.1 Objectives 1](#_Toc129643809)

[1.1.1 General Objective 1](#_Toc129643810)

[1.2 Theory 1](#_Toc129643811)

[1.2.1 Volume of Cube 1](#_Toc129643812)

[1.2.2 Volume of Cylinder 2](#_Toc129643813)

[1.2.3 Volume of Cone 2](#_Toc129643814)

[1.2.4 Volume of Sphere 3](#_Toc129643815)

[1.3 Syntax and functions 3](#_Toc129643816)

[1.3.1 Input function 3](#_Toc129643817)

[1.3.2 Print function 3](#_Toc129643818)

[1.4 Methodology 3](#_Toc129643819)

[1.5 Results and Discussion 4](#_Toc129643820)

[1.6 Conclusion 5](#_Toc129643821)

[2.1 Objectives 6](#_Toc129643824)

[2.1.1 General Objective 6](#_Toc129643825)

[2.2 Theory 6](#_Toc129643826)

[2.2.1 Tax Table 6](#_Toc129643827)

[2.3 Syntax and Functions 6](#_Toc129643828)

[2.3.1 Input Function 6](#_Toc129643829)

[2.3.2 Print Function 7](#_Toc129643830)

[2.3.3 Functions 7](#_Toc129643831)

[2.4 Methodology 7](#_Toc129643832)

[2.5 Results and Discussion 8](#_Toc129643833)

[2.6 Conclusion 8](#_Toc129643839)

[3.1 Objectives 9](#_Toc129643841)

[3.1.1 General Objective 9](#_Toc129643842)

[3.2 Theory 9](#_Toc129643843)

[3.2.1 Guessing Game 9](#_Toc129643844)

[3.3 Syntax and Functions 9](#_Toc129643845)

[3.3.1 Input Function 9](#_Toc129643846)

[3.3.2 Print Function 10](#_Toc129643847)

[3.3.3 Import Function 10](#_Toc129643848)

[3.3.4 If-Else Function 10](#_Toc129643849)

[3.4 Methodology 10](#_Toc129643850)

[3.5 Results and Discussion 11](#_Toc129643851)

[3.6 Conclusion 11](#_Toc129643857)

[4.1 Objectives 12](#_Toc129643859)

[4.1.1 General Objective 12](#_Toc129643860)

[4.2 Theory 12](#_Toc129643861)

[4.2.1 Celsius 12](#_Toc129643862)

[4.2.2 Fahrenheit 12](#_Toc129643863)

[4.2.3 Kelvin 13](#_Toc129643864)

[4.3 Syntax and Functions 13](#_Toc129643865)

[4.3.1 Input Function 13](#_Toc129643866)

[4.3.2 Print Function 13](#_Toc129643867)

[4.3.3 Split Function 13](#_Toc129643868)

[4.3.4 Import Function 14](#_Toc129643869)

[4.3.5 If-Else Function 14](#_Toc129643870)

[4.4 Methodology 14](#_Toc129643871)

[4.5 Results and Discussion 15](#_Toc129643872)

[4.6 Conclusion 16](#_Toc129643878)

[5.1 Objectives 17](#_Toc129643880)

[5.1.1 General Objective 17](#_Toc129643881)

[5.2 Theory 17](#_Toc129643882)

[5.2.1 Sum of All Numbers 17](#_Toc129643883)

[5.3 Syntax and Functions 18](#_Toc129643884)

[5.3.1 Input Function 18](#_Toc129643885)

[5.3.2 Print Function 18](#_Toc129643886)

[5.3.3 Import Function 18](#_Toc129643887)

[5.3.4 For Loop Function 18](#_Toc129643888)

[5.3.5 While Loop Function 18](#_Toc129643889)

[5.4 Methodology 19](#_Toc129643890)

[5.5 Results and Discussion 20](#_Toc129643891)

[5.6 Conclusion 20](#_Toc129643897)

[References (Last section in the file. Include if applicable) 21](#_Toc129643898)

# List of Figures

**Experiment No. 1: Basic Syntax, Variables, and Arithmetic**

[Figure 1. 1 Volume of a Cube Formula 1](file:///C:\Users\Eliza\OneDrive%20-%20De%20La%20Salle%20University%20-%20Manila\Desktop\DLSU\2ND%20TERM\LBYCPA\DE%20LA%20ROSA_BONUS%20ACT%202.docx#_Toc129573775)

[Figure 1. 2 Volume of Cylinder Formula 2](file:///C:\Users\Eliza\OneDrive%20-%20De%20La%20Salle%20University%20-%20Manila\Desktop\DLSU\2ND%20TERM\LBYCPA\DE%20LA%20ROSA_BONUS%20ACT%202.docx#_Toc129573776)

[Figure 1. 3 Volume of a Cone Formula 2](file:///C:\Users\Eliza\OneDrive%20-%20De%20La%20Salle%20University%20-%20Manila\Desktop\DLSU\2ND%20TERM\LBYCPA\DE%20LA%20ROSA_BONUS%20ACT%202.docx#_Toc129573777)

[Figure 1. 4 Volume of a Sphere Formula 3](file:///C:\Users\Eliza\OneDrive%20-%20De%20La%20Salle%20University%20-%20Manila\Desktop\DLSU\2ND%20TERM\LBYCPA\DE%20LA%20ROSA_BONUS%20ACT%202.docx#_Toc129573778)

**Experiment No. 2: Tax Calculator**

[Figure 2. 1 Tax Table 6](file:///C:\Users\Eliza\OneDrive%20-%20De%20La%20Salle%20University%20-%20Manila\Desktop\DLSU\2ND%20TERM\LBYCPA\DE%20LA%20ROSA_BONUS%20ACT%202.docx#_Toc129573846)

**Experiment No. 3: Guessing Game**

[Figure 3. 1 Example of a Guessing Game 9](file:///C:\Users\Eliza\OneDrive%20-%20De%20La%20Salle%20University%20-%20Manila\Desktop\DLSU\2ND%20TERM\LBYCPA\DE%20LA%20ROSA_BONUS%20ACT%202.docx#_Toc129573763)

**Experiment No. 4: Temperature Converter**

[Figure 4. 1 Celsius Conversion Formulas 12](file:///C:\Users\Eliza\OneDrive%20-%20De%20La%20Salle%20University%20-%20Manila\Desktop\DLSU\2ND%20TERM\LBYCPA\DE%20LA%20ROSA_BONUS%20ACT%202.docx#_Toc129573767)

[Figure 4. 2 Fahrenheit Conversion Formulas 12](file:///C:\Users\Eliza\OneDrive%20-%20De%20La%20Salle%20University%20-%20Manila\Desktop\DLSU\2ND%20TERM\LBYCPA\DE%20LA%20ROSA_BONUS%20ACT%202.docx#_Toc129573768)

[Figure 4. 3 Kelvin Conversion Formulas 13](file:///C:\Users\Eliza\OneDrive%20-%20De%20La%20Salle%20University%20-%20Manila\Desktop\DLSU\2ND%20TERM\LBYCPA\DE%20LA%20ROSA_BONUS%20ACT%202.docx#_Toc129573769)

**Experiment No. 5: Python Loop Performance**

[Figure 5. 1 Example of Sum of All Numbers 17](file:///C:\Users\Eliza\OneDrive%20-%20De%20La%20Salle%20University%20-%20Manila\Desktop\DLSU\2ND%20TERM\LBYCPA\DE%20LA%20ROSA_BONUS%20ACTIVITY.docx#_Toc129643899)

# List of Tables

**Experiment No. 1: Basic Syntax, Variables, and Arithmetic**

[Table 1. 1 Summary of results for the Volume of a Cube 4](#_Toc126794716)

[Table 1. 2 Summary of results for the Volume of a Cylinder 4](#_Toc126794717)

[Table 1. 3 Summary of results for the Volume of a Cone 5](#_Toc126794718)

Table 1. 4 Summary of results for the Volume of a Sphere 5

**Experiment No. 2: Tax Calculator**

[Table 2. 1 Summary of results for computing for Monthly and Annual Tax 7](#_Toc126794999)

**Experiment No. 3: Guessing Game**

[Table 3. 1 Summary of results in conducting Guessing Game 11](#_Toc129643904)

**Experiment No. 4: Temperature Converter**

[Table 4. 1 Summary of results of converting the Celsius Unit 15](#_Toc129643927)

[Table 4. 2 Summary of results of converting the Fahrenheit Unit 16](#_Toc129643928)

[Table 4. 3 Summary of results of converting the Kelvin Unit 16](#_Toc129643929)

**Experiment No. 5: Python Loop Performance**

[Table 5. 1 Summary of results of Sum of All Numbers 20](#_Toc129643941)

**Experiment No. 1: Basic Syntax, Variables, and Arithmetic**

## Objectives

### General Objective

To demonstrate the use of python to solve for the volume of cube, cylinder, cone, and sphere.

## Theory

### Volume of Cube

Graphical user interface, application

Description automatically generatedA solid three-dimensional figure with six square faces or sides is known as a cube. The total amount of space a cube takes up is its volume. The cube's square shape means that all its faces are square, and as a result, all its edges will be of equal length. As a result, the cube's length, width, and height are all equal. If a cube's length, width, and height are equal to "a," then:

Figure 1. Volume of a Cube Formula

### Volume of Cylinder

**Graphical user interface

Description automatically generated**The cylinder has a circular base and is a three-dimensional shape. The density of a cylinder is determined by its volume, which represents how much material can be immersed in it or carried inside of it. We determine the area occupied by each disk separately, add them together, and then determine the area occupied by a cylinder. As a result, the product of the base area and height can be used to determine the cylinder's volume.

Figure 1. Volume of Cylinder Formula

### Volume of Cone

Graphical user interface

Description automatically generatedA cone's circular base tapers from a flat base to a point known as the apex or vertex in three dimensions. The area or capacity of a cone is determined by its volume. If the height and radius of a cone are known, a formula can be used to determine the volume of the cone.

Figure 1. Volume of a Cone Formula

### Volume of Sphere

Graphical user interface

Description automatically generatedThe capacity of a sphere is its volume. It is the area that the sphere occupies. Cubic measurements of a sphere's volume include m3, cm3, in3, etc. The sphere has a round, three-dimensional shape. Its shape is determined by three axes: the x, y, and z axes. Since the cross-section of the sphere is a circle, the volume in this case is dependent on the diameter of the sphere's radius.

Figure 1. Volume of a Sphere Formula

## Syntax and functions

### Input function

**Usage:** Allows user to input variables.

**Syntax:**

input(prompt)

### Print function

**Usage:** Display words/sentences for user to read.

**Syntax:**

print(“example”)

## Methodology

print ("Computation of Volume: Cube, Cylinder, Cone, Sphere")

print ("\nSolving for the volume of a Cube")

ECube = float(input("Enter edge of the Cube: "))

VCube = ECube\*\*3 #Formula for Volume of Cube

print ("\nThe volume of the Cube is {}".format(VCube))

print ("\nSolving for the volume of a Cylinder")

RCyl = float(input("Enter radius of the Cylinder: "))

HCyl= float(input("Enter height of the Cylinder: "))

VCyl = 3.14\*RCyl\*\*2\*HCyl #Formula for Volume of Cylinder

print ("\nThe volume of the Cylinder is {}".format(VCyl))

print ("\nSolving for the volume of a Cone")

RCone = float(input("Enter radius of the Cone: "))

HCone= float(input("Enter height of the Cone: "))

VCone= 3.14\*RCone\*\*2\*HCone/3 #Formula for Volume of Cone

print ("\nThe volume of the Cone is {}".format(VCone))

print ("\nSolving for the volume of a Sphere")

RSphere = float(input("Enter radius of the Sphere: "))

VSphere = 4/3\*3.14\*RSphere\*\*3 #Formula for Volume of Sphere

print ("\nThe volume of the Cube is {}".format(VSphere))

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

## Results and Discussion

Table 1. Summary of results for the Volume of a Cube

|  |  |  |
| --- | --- | --- |
| **Dimensions** | **Manual**  **Computation** | **Python**  **Code** |
| Edge |
| 5 | 125 | 125 |
| 3 | 27 | 27 |

Table 1. Summary of results for the Volume of a Cylinder

|  |  |  |  |
| --- | --- | --- | --- |
| **Dimensions** | | **Manual**  **Computation** | **Python**  **Code** |
| **Radius** | **Height** |
| 2 | 4 | 50.24 | 50.24 |
| 6 | 8 | 904.32 | 904.32 |

Table 1. Summary of results for the Volume of a Cone

|  |  |  |  |
| --- | --- | --- | --- |
| **Dimensions** | | **Manual**  **Computation** | **Python**  **Code** |
| **Radius** | **Height** |
| 2 | 4 | 16.74666667 | 16.746666666666666 |
| 6 | 8 | 301.44 | 301.44 |

Table 1. Summary of results for the Volume of a Sphere

|  |  |  |
| --- | --- | --- |
| **Dimensions** | **Manual**  **Computation** | **Python**  **Code** |
| Radius |
| 5 | 523.3333333 | 523.3333333333334 |
| 3 | 113.04 | 113.03999999999999 |

## Conclusion

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

**Experiment No. 2: Tax Calculator**



## Objectives

### General Objective

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

## Theory

### Tax Table

Table

Description automatically generatedTRAIN, 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:

Figure 2. Tax Table

## Syntax and Functions

### Input Function

**Usage:** Allows user to input variables.

**Syntax:**

input(prompt)

### Print Function

**Usage:** Display words/sentences for user to read.

**Syntax:**

print(“example”)

### 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”():

## 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 #35% Tax Rate

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")

## Results and Discussion

Table 2. 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 |



## Conclusion

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.

**Experiment No. 3: Guessing Game**



## Objectives

### General Objective

To demonstrate the use of python to generate random integers and have the user guess the number given 10 chances.

## Theory

### Guessing Game

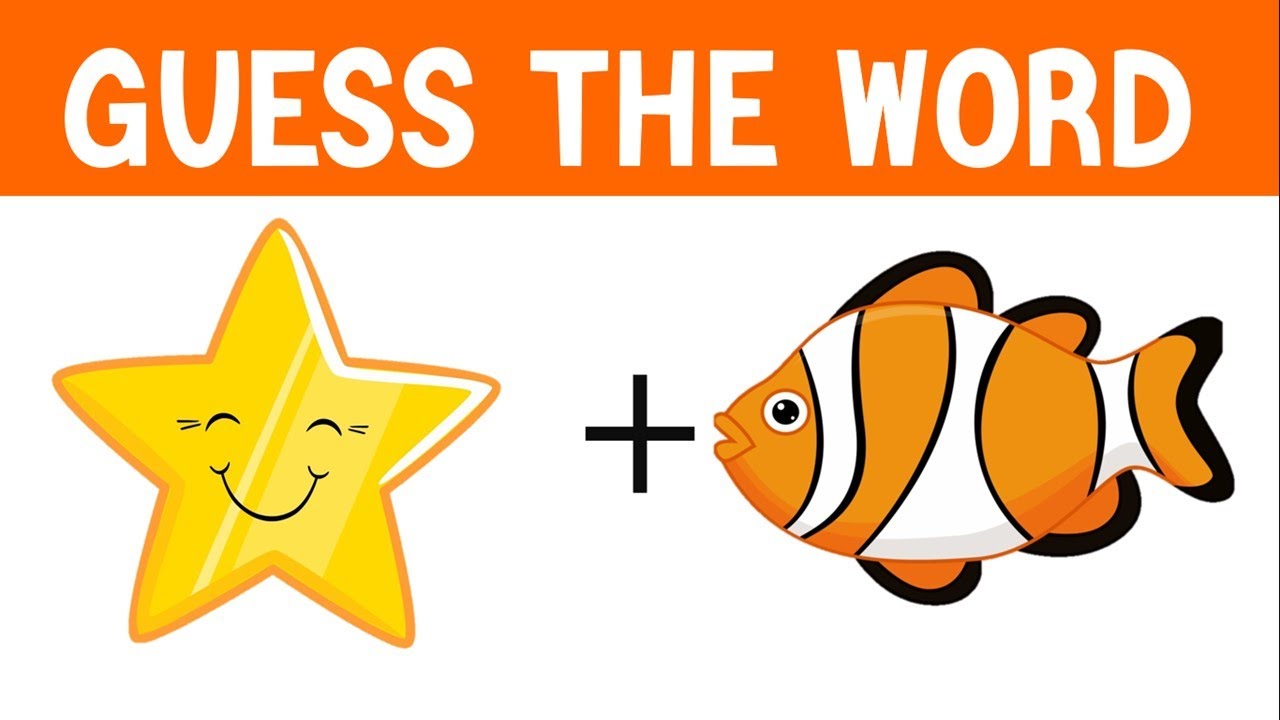
A guessing game is a game in which the player has to correctly guess the answer.

Figure 3. Example of a Guessing Game

## Syntax and Functions

### Input Function

**Usage:** Allows user to input variables

**Syntax:**

input(prompt)

### Print Function

**Usage:** Display words/sentences for user to read.

**Syntax:**

print(“example”)

### Import Function

**Usage:** Makes code in one module available in another.

**Syntax:**

import os

### If-Else Function

**Usage:** Executes a block of code if a specified condition is true. If the condition is false, another block of code can be executed.

**Syntax:**

if example == example1

print “equal”

elif example > example1

print “greater than”

else

print “try”

## Methodology

import random

import os

number = random.randint(1, 100)

trials = 10

print ("Guessing Game!")

for count in range(trials):

guess = int(input("Guess the number between 1 and 100: "))

if guess == number:

print("\nCongratulations!\nYou guessed the correct number in", count+1, "trials.\n")

break

elif guess < number:

print("Your guess is too low. Try again.")

elif guess > number:

print("Your guess is too high. Try again.")

else:

print("\nSorry, you have used all", trials, "trials. The correct number was", number)

os.system("pause")

## Results and Discussion

Table 3. Summary of results in conducting Guessing Game

|  |  |
| --- | --- |
| **Random Integer** | **User’s Guess (10 Chances)** |
| 27 | 5, 10, 50, 40, 30, 20, 25, 26, 27 |
| 3 | 100, 50, 40, 30, 10, 1, 2, 3 |



## Conclusion

From the results above, it can be concluded that the program created is working correctly. Although, manual computations cannot be exuceted with this program as this is a guessing game, where the integer given is random.

**Experiment No. 4: Temperature Converter**



## Objectives

### General Objective

To demonstrate the use of python to convert the unit of a temperature.

## Theory

### Celsius

Table

Description automatically generatedCelsius is a scale for measuring temperature, in which water freezes at 0 degrees and boils at 100 degrees. It is represented by the symbol ° C.

Figure 4. Celsius Conversion Formulas

### Fahrenheit

Graphical user interface, application

Description automatically generatedTable

Description automatically generatedFahrenheit is a scale for measuring temperature, in which water freezes at 32 degrees and boils at 212 degrees. It is represented by the symbol ° F.

Figure 4. Fahrenheit Conversion Formulas

### Kelvin

Graphical user interface, application, Word

Description automatically generatedTable

Description automatically generatedKelvin is the basic SI unit of thermodynamic temperature; the fraction 1⁄273.16 of the thermodynamic temperature of the triple point of water. It is represented by the symbol K.

Figure 4. Kelvin Conversion Formulas

## Syntax and Functions

### Input Function

**Usage:** Allows user to input variables

**Syntax:**

input(prompt)

### Print Function

**Usage:** Display words/sentences for user to read.

**Syntax:**

print(“example”)

### Split Function

**Usage:** Breaks character data into segments based on separators such as spaces or commas and returns a specified segment.

**Syntax:**

ex = sample.split(“ “)

### Import Function

**Usage:** Makes code in one module available in another.

**Syntax:**

import os

### If-Else Function

**Usage:** Executes a block of code if a specified condition is true. If the condition is false, another block of code can be executed.

**Syntax:**

if example == example1

print “equal”

elif example > example1

print “greater than”

else

print “try”

## Methodology

import os

import script\_1

value = input("Input a temperature: ")

x = value.split(" ")

number = float(x[0])

unit = str(x[1])

if unit == 'C':

print("1. Convert to Fahrenheit")

print("2. Convert to Kelvin")

option\_1 = input("\nInput desired program: ")

if option\_1 == "1":

converted = script\_1.C\_to\_F(number)

print("\n{} C IS {} F".format(number, converted))

elif option\_1 == "2":

converted = script\_1.C\_to\_K(number)

print("\n{} C IS {} K".format(number, converted))

else:

print("Invalid input. Try again.")

elif unit == 'K':

print("1. Convert to Celsius")

print("2. Convert to Fahrenheit")

option\_2 = input("\nInput desired program: ")

if option\_2 == "1":

converted = script\_1.K\_to\_C(number)

print("\n{} K IS {} C".format(number, converted))

elif option\_2 == "2":

converted = script\_1.K\_to\_F(number)

print("\n{} K IS {} F".format(number, converted))

else:

print("Invalid input. Try again.")

elif unit == 'F':

print("1. Convert to Celsius")

print("2. Convert to Kelvin")

option\_3 = input("\nInput desired program: ")

if option\_3 == "1":

converted = script\_1.F\_to\_C(number)

print("\n{} F IS {} C".format(number, converted))

elif option\_3 == "2":

converted = script\_1.F\_to\_K(number)

print("\n{} F IS {} K".format(number, converted))

else:

print("Invalid input. Try again.")

else:

print("Please input a proper unit")

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

os.system("pause")

## Results and Discussion

|  |  |  |  |
| --- | --- | --- | --- |
| **Temperature with Unit** | **Conversion** | **Manual Computation** | **Python**  **Code** |
| 25 C | F | 77 | 77.0 F |
| 30 C | K | 303.15 | 303.15 K |

Table 4. Summary of results of converting the Celsius Unit

Table 4. Summary of results of converting the Fahrenheit Unit



|  |  |  |  |
| --- | --- | --- | --- |
| **Temperature with Unit** | **Conversion** | **Manual Computation** | **Python**  **Code** |
| 50 F | C | 10 | 10.0 C |
| 75 F | K | 297.0388889 | 297.0388888888889 K |

Table 4. Summary of results of converting the Kelvin Unit

|  |  |  |  |
| --- | --- | --- | --- |
| **Temperature with Unit** | **Conversion** | **Manual Computation** | **Python Code** |
| 300 K | C | 26.85 | 26.850000000000023 C |
| 275 K | F | 35.33 | 35.33000000000004 F |

## Conclusion

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

**Experiment No. 5: Python Loop Performance**



## Objectives

### General Objective

To demonstrate the use of python to calculate the sum of all numbers from 1 to n while importing the time module.

## Theory

### Sum of All Numbers

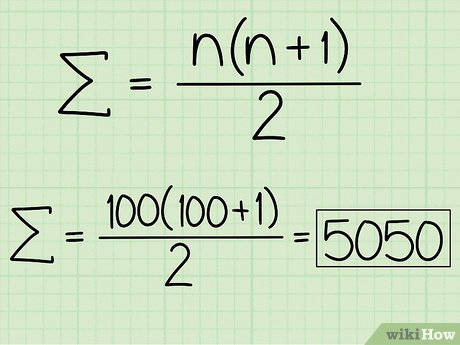
The sum brings two or more numbers together to make a new total. The sum of all numbers is putting all the numbers from a certain range together. An example of this is 10, to get the sum of all the numbers of 10, you add together the numbers: 1, 2, 3, 4, 5, 6, 7, 8, 8, 10. With that, the sum of all the numbers of 10 is 55.

Figure 5. Example of Sum of All Numbers

## Syntax and Functions

### Input Function

**Usage:** Allows user to input variables

**Syntax:**

input(prompt)

### Print Function

**Usage:** Display words/sentences for user to read.

**Syntax:**

print(“example”)

### Import Function

**Usage:** Makes code in one module available in another.

**Syntax:**

import os

### For Loop Function

**Usage:** Iterates over a sequence.

**Syntax:**

example = 0

for count in range (1, input+1):

example = example + 1

return example

### While Loop Function

**Usage:** Repeats a specific block of code an unknown number of times, until a condition is met.

**Syntax:**

example = 0

count = 0

while count <= input

example = example + count

count = count + 1

return example

## Methodology

import os

import time

def for\_loop(n):

sum = 0

for i in range(1, n+1):

sum = sum + i

return sum

def while\_loop(n):

sum = 0

i = 1

while i <= n:

sum = sum + i

i = i + 1

return sum

n\_values = [10, 100, 1000, 10000, 100000]

for n in n\_values:

start\_time = time.time()

result = for\_loop(n)

end\_time = time.time()

print(f"For loop sum of {n} numbers: {result}, execution time: {end\_time - start\_time:.6f} seconds")

start\_time = time.time()

result = while\_loop(n)

end\_time = time.time()

print(f"While loop sum of {n} numbers: {result}, execution time: {end\_time - start\_time:.6f} seconds")

print("=" \* 78)

os.system("pause")

## Results and Discussion

Table 5. Summary of results of Sum of All Numbers

|  |  |  |
| --- | --- | --- |
| **Number** | **Manual Computation** | **Python Code** |
| 10 | 55 | 55 |
| 100 | 5050 | 5050 |



## Conclusion

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

# References (Last section in the file. Include if applicable)

Include references in IEEE Format