Artificial Intelligence

CSL 411

Lab Journal

Ahmad Hassan
01-134191-002
BSCS 6B

Department of Computer Science
BAHRIA UNIVERSITY, ISLAMABAD

Lab # 1: Introduction to Python

Objectives:

To be able to install python. Introduction to python IDLE and learning to code in python

Tools Used:

IDLE (Python 3.4 GUI Python)

Submission Date:

Evaluation: **Signatures of Lab Engineer:**

Task # 1:

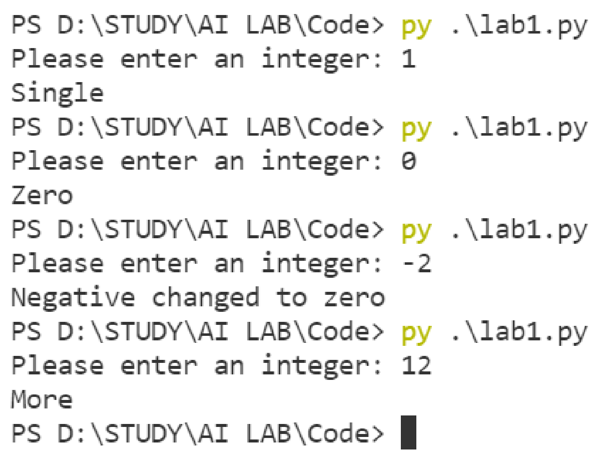
Open IDLE and run the following program. Try different integer values for separate runs of the program. Play around with the indentation of the program lines of code and run it again. See what happens. Make a note of what changes you made and how it made the program behave. Also note any errors, as well as the changes you need to make to remove the errors.

```
x = input("Please enter an integer: ")

if x < 0:

    x = 0

    print('Negative changed to zero')

elif x == 0:

    print('Zero')

elif x == 1:

    print('Single')

else:

    print('More')
```

Procedure/Program:

```
x = int(input("Please enter an integer: "))
if x < 0:
    x = 0
    print('Negative changed to zero')
elif x == 0:
    print('Zero')
elif x == 1:
    print('Single')
else:
    print('More')
```

Result/Output:



```
PS D:\STUDY\AI LAB\Code> py .\lab1.py
Please enter an integer: 1
Single
PS D:\STUDY\AI LAB\Code> py .\lab1.py
Please enter an integer: 0
Zero
PS D:\STUDY\AI LAB\Code> py .\lab1.py
Please enter an integer: -2
Negative changed to zero
PS D:\STUDY\AI LAB\Code> py .\lab1.py
Please enter an integer: 12
More
PS D:\STUDY\AI LAB\Code>
```

Analysis/Conclusion:

There was an error because the input function takes value as a string and to deal with this problem we are using the method of casting `x = int(input("Please enter an integer: "))`

Task # 2:

1. Write a simple unit calculator program. Follow the steps below:
   a. Declare and define a function named Menu which displays a list of choices for user such as meter to kilometer, kilometer to meter, centimeter to meter, & centimeter to millimeter. It takes the choice from user as an input and return.
   b. Define and declare a separate function for each choice.

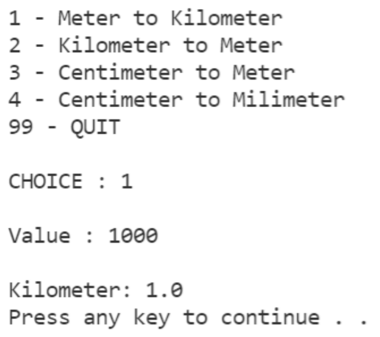
In the main body of the program call respective function depending on user's choice.

   c. Program should not terminate till user chooses option to "Quit".

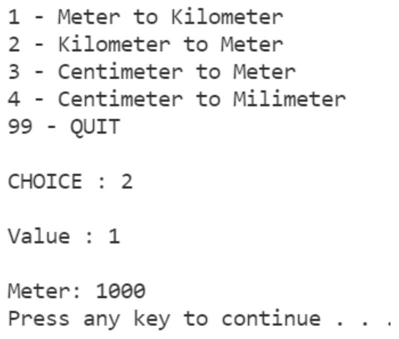
Procedure/Program:

```
import os


def menu():
    # print("1 - Meter to Kilometer\n2 - Kilometer to
Meter\n3 - Centimeter to Meter\n4 - Centimeter to
Milimeter\nCHOICE:")
    choice = int(input(
        "1 - Meter to Kilometer\n2 - Kilometer to
Meter\n3 - Centimeter to Meter\n4 - Centimeter to
Milimeter\n99 - QUIT\n\nCHOICE : "))
    return choice


def meterToKilometer(data):
    return data/1000

# kilometer to meter


def kilometerToMeter(data):
    return data/1000

# cewntimeter to meter


def centimeterToMeter(data):
    return data/100

# centimeter to milimeter

def centimeterToMillimeter(data):
    return data*10
# MAIN


choice = 0
while 1:
    os.system('cls')
    choice = menu()
    if choice == 99:
        print("\nAllah Hafiz!")
        break
    data = int(input("\nValue : "))
    if choice == 1:
        print(f"\nKilometer:
{meterToKilometer(data)}")
    elif choice == 2:
        print(f"\nMeter: {kilometerToMeter(data)}")
    elif choice == 3:
        print(f"\nMeter:
{centimeterToMeter(data)}\n")
    elif choice == 4:
        print(f"\nMillimeter:
{centimeterToMillimeter(data)}\n")
    else:
        print('\nINVALID CHOICE!')
    os.system('pause')


print('\n\nWhile Ends Here!')
```

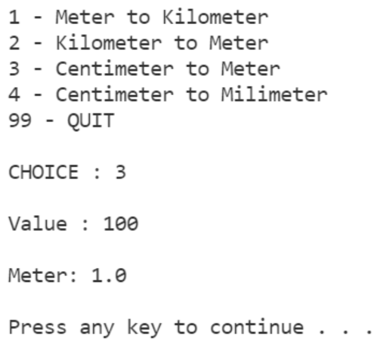
Result/Output:



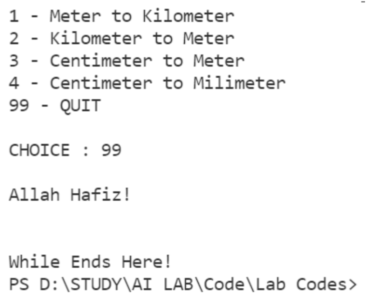
```
1 - Meter to Kilometer
2 - Kilometer to Meter
3 - Centimeter to Meter
4 - Centimeter to Milimeter
99 - QUIT

CHOICE : 1

Value : 1000

Kilometer: 1.0
Press any key to continue . .
```



```
1 - Meter to Kilometer
2 - Kilometer to Meter
3 - Centimeter to Meter
4 - Centimeter to Milimeter
99 - QUIT

CHOICE : 2

Value : 1

Meter: 1000
Press any key to continue . . .
```



```
1 - Meter to Kilometer
2 - Kilometer to Meter
3 - Centimeter to Meter
4 - Centimeter to Milimeter
99 - QUIT

CHOICE : 3

Value : 100

Meter: 1.0

Press any key to continue . . .
```



```
1 - Meter to Kilometer
2 - Kilometer to Meter
3 - Centimeter to Meter
4 - Centimeter to Milimeter
99 - QUIT

CHOICE : 99

Allah Hafiz!


While Ends Here!
PS D:\STUDY\AI LAB\Code\Lab Codes>
```

Analysis/Conclusion:

Task # 3:

1. Create a class name basic_calc with following attributes and methods;
   Two integers (values are passed with instance creation)
   Different methods such as addition, subtraction, division, multiplication
   Create another class inherited from basic_calc named s_calc which should have the following additional methods;
   Factorial, x_power_y, log, ln etc

2. Modify the classes created in the above task under as follows:
   Create a module name basic.py having the class name basic_calc with all the attributes and methods defined before.
   Now import the basic.py module in your program and do the inheritance step defined before i.e. Create another class inherited from basic_calc named s_calc which should have the following additional methods;
   Factorial, x_power_y, log, ln etc

Procedure/Program:

Part 1:

```
import os
import math as solve

class basic_calc:
    def __init__(self, val1, val2):
        self.val1 = val1
        self.val2 = val2
    def sum(self):
        return self.val1+self.val2
    def subt(self):
        return self.val1-self.val2
    def div(self):
        return self.val1/self.val2
    def prod(self):
        return self.val1*self.val2
    def __str__(self):
        return (f"{self.val1} &
{self.val2}")

class s_calc(basic_calc):
    def fact(self):
        num1 = self.val1
        num2 = self.val2
        val1 = val2 = 1
        while 1:
            val1 *= num1
            num1 -= 1
            if num1 == 0:
                break
        while 1:
            val2 *= num2
            num2 -= 1
            if num2 == 0:
                break
        return s_calc(val1, val2)
    def power(self):
        x = self.val1
        y = self.val2
        x_power_y = x**y
        y_power_x = y**x
        return s_calc(x_power_y,
y_power_x)
    def log(self):
        x = self.val1
        y = self.val2
        log_x_base_y = solve.log(x, y)
        log_y_base_x = solve.log(y, x)
        return s_calc(log_x_base_y,
log_y_base_x)
    def ln(self):
        x = self.val1
        y = self.val2
        ln_x = solve.log(x)
        ln_y = solve.log(y)
        return s_calc(ln_x, ln_y)

bcalc = basic_calc(10, 5)
scalc = s_calc(bcalc.val1, bcalc.val2)
print("\nbasic_calc Starts Here!\n")
print(f"Addition   of {bcalc} =
{bcalc.sum()}")
print(f"Difference of {bcalc} =
{bcalc.subt()}")
print(f"Division   of {bcalc} =
{bcalc.div()}")
print(f"Product    of {bcalc} =
{bcalc.prod()}")
print("\ns_calc Starts Here!\n")
print(f"Product of {scalc} =
{scalc.fact()}")
print(f"Power   of {scalc} =
{scalc.power()}")
print(f"Log     of {scalc} =
{scalc.log()}")
print(f"ln      of {scalc} =
{scalc.ln()}")
input()
```

Part 2:

basic.py

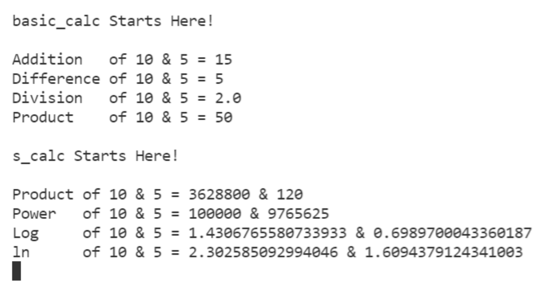
```python
import math as solve

class basic_calc:
    def __init__(self, val1, val2):
        self.val1 = val1
        self.val2 = val2
    def sum(self):
        return self.val1+self.val2
    def subt(self):
        return self.val1-self.val2
    def div(self):
        return self.val1/self.val2
    def prod(self):
        return self.val1*self.val2
    def __str__(self):
        return (f"{self.val1} & 
{self.val2}")
```

lab_1_task3.py

```python
import math as solve
import basic as calc
class s_calc(calc.basic_calc):
    def fact(self):
        num1 = self.val1
        num2 = self.val2
        val1 = val2 = 1
        while 1:
            val1 *= num1
            num1 -= 1
            if num1 == 0:
                break
        while 1:
            val2 *= num2
            num2 -= 1
            if num2 == 0:
                break
        return s_calc(val1, val2)
    def power(self):
        x = self.val1
        y = self.val2
        x_power_y = x**y
        y_power_x = y**x
        return s_calc(x_power_y, 
y_power_x)
    def log(self):
        x = self.val1
        y = self.val2
        log_x_base_y = solve.log(x, y)
        log_y_base_x = solve.log(y, x)
        return s_calc(log_x_base_y, 
log_y_base_x)
    def ln(self):
        x = self.val1
        y = self.val2
        ln_x = solve.log(x)
        ln_y = solve.log(y)
        return s_calc(ln_x, ln_y)

bcalc = calc.basic_calc(10, 5)
scalc = s_calc(bcalc.val1, bcalc.val2)
print("\nbasic_calc Starts Here!\n")
print(f"Addition   of {bcalc} = 
{bcalc.sum()}")
print(f"Difference of {bcalc} = 
{bcalc.subt()}")
print(f"Division   of {bcalc} = 
{bcalc.div()}")
print(f"Product    of {bcalc} = 
{bcalc.prod()}")
print("\ns_calc Starts Here!\n")
print(f"Product of {scalc} = 
{scalc.fact()}")
print(f"Power   of {scalc} = 
{scalc.power()}")
print(f"Log     of {scalc} = 
{scalc.log()}")
print(f"ln      of {scalc} = 
{scalc.ln()}")
input()
```

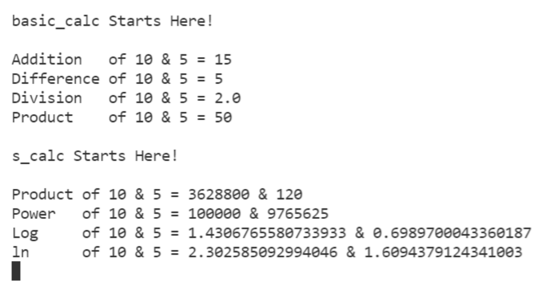
Result/Output:

Part 1



```
basic_calc Starts Here!

Addition   of 10 & 5 = 15
Difference of 10 & 5 = 5
Division   of 10 & 5 = 2.0
Product    of 10 & 5 = 50

s_calc Starts Here!

Product of 10 & 5 = 3628800 & 120
Power   of 10 & 5 = 100000 & 9765625
Log     of 10 & 5 = 1.4306765580733933 & 0.6989700043360187
ln      of 10 & 5 = 2.302585092994046 & 1.6094379124341003
```

Part 2



```
basic_calc Starts Here!

Addition   of 10 & 5 = 15
Difference of 10 & 5 = 5
Division   of 10 & 5 = 2.0
Product    of 10 & 5 = 50

s_calc Starts Here!

Product of 10 & 5 = 3628800 & 120
Power   of 10 & 5 = 100000 & 9765625
Log     of 10 & 5 = 1.4306765580733933 & 0.6989700043360187
ln      of 10 & 5 = 2.302585092994046 & 1.6094379124341003
```

Analysis/Conclusion:

No change found in output.