***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:** | **Result/Output:** |
| 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') |  |

**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:
   1. 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.
   2. Define and declare a separate function for each choice.

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

* 1. 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:**

Text

Description automatically generatedText

Description automatically generatedA screenshot of a computer

Description automatically generated with low confidence

Text

Description automatically generated

**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

1. 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

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

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**

A picture containing text

Description automatically generated

**Part 2**

A picture containing text

Description automatically generated

**Analysis/Conclusion:**

No change found in output.