

# *Artificial Intelligence*

CSL 411

## *Lab Journal*



**Student Name:**

**Enrolment No.:**

**Class and Section:**

**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 = 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:

```
In [91]: runfile('C:/Users/Student/file.py', wdir='C:/Users/Student')
```

```
Please enter an integer: -1
Negative changed to zero
```

```
In [92]:
```

---

### Analysis/Conclusion:

Input function by default takes string so for entering int value we have to do type casting

## 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.
  - c. In the main body of the program call respective function depending on user's choice.
  - d. Program should not terminate till user chooses option to "Quit".

### Procedure/Program:

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
Created on Wed Mar 3 10:57:42 2021
```

```
@author: Student
```

```
"""
```

```
def MeterToKilometer():
```

```
    num = float(input("Enter Meters:"))
```

```
    num = num/1000
```

```
    print("Meter to Kilometer: ", num,"KM")
```

```
def KilometerToMeter():
```

```
    num = float(input("Enter Kilometers:"))
```

```
    num = num*1000
```

```
    print("Kilometer to Meter: ", num,"M")
```

```
def CentimeterToMeter():
```

```
    num = float(input("Enter Centimeters:"))
```

```
    num = num/100
```

```
    print("Centimeter to Meter: ", num,"M")
```

```
def CentimeterToMillimeter():
```

```
    num = float(input("Enter Centimeters:"))
```

```
    num = num*10
```

```
    print("Centimeter to Millimeter: ", num,"MM")
```

```
def Menu():
```

```
    print("\tMenu")
```

```
    print("1)Meter to Kilometer")
```

```
    print("2)Kilometer to Meter")
```

```
    print("3)Centimeter to Meter")
```

```
    print("4)Centimeter to Millimeter")
```

```
    print("5)Quit")
```

```
    x = int(input("Enter Your Choice: "))
```

```
    return x
```

```
while 1:
```

```
x = Menu()
if x == 1:
    MeterToKilometer()
elif x == 2:
    KilometerToMeter()
elif x == 3:
    CentimeterToMeter()
elif x == 4:
    CentimeterToMilimeter()
elif x == 5:
    break
else:
    print("Invalid Choice, Try Again\n")
```

**Result/Output:**

```
In [97]: runfile('C:/Users/Student/Task2.py', wdir='C:/Users/Student')
```

```
Menu
```

- 1)Meter to Kilometer
- 2)Kilometer to Meter
- 3)Centimeter to Meter
- 4)Centimeter to Millimeter
- 5)Quit

```
Enter Your Choice: 1
```

```
Enter Meters:5
```

```
Meter to Kilometer: 0.005 KM
```

```
Menu
```

- 1)Meter to Kilometer
- 2)Kilometer to Meter
- 3)Centimeter to Meter
- 4)Centimeter to Millimeter
- 5)Quit

```
Enter Your Choice: 2
```

```
Enter Kilometers:0.005
```

```
Kilometer to Meter: 5.0 M
```

```
Menu
```

- 1)Meter to Kilometer
- 2)Kilometer to Meter
- 3)Centimeter to Meter
- 4)Centimeter to Millimeter
- 5)Quit

```
Enter Your Choice: 3
```

```
Enter Centimeters:5
```

```
Centimeter to Meter: 0.05 M
```

```
Menu
```

- 1)Meter to Kilometer
- 2)Kilometer to Meter
- 3)Centimeter to Meter

```
4)Centimeter to Millimeter
5)Quit
```

Enter Your Choice: 3

```
Enter Centimeters:5
Centimeter to Meter:  0.05 M
    Menu
```

```
1)Meter to Kilometer
2)Kilometer to Meter
3)Centimeter to Meter
4)Centimeter to Millimeter
5)Quit
```

Enter Your Choice: 4

```
Enter Centimeters:5
Centimeter to Millimeter:  50.0 MM
    Menu
```

```
1)Meter to Kilometer
2)Kilometer to Meter
3)Centimeter to Meter
4)Centimeter to Millimeter
5)Quit
```

Enter Your Choice: 5

In [98]: |

**Analysis/Conclusion:** The program will not end until user it self quits it.

### 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

### Procedure/Program:

```
# -*- coding: utf-8 -*-
"""
```

Created on Wed Mar 3 08:39:27 2021

```
@author: Student
"""
```

```
import math
class basic_calc():
    def __init__(self,x,y):
```

```

        self.x=x
        self.y=y
    def Sum(self):
        return self.x+self.y
    def Sub(self):
        return self.x-self.y
    def Mul(self):
        return self.x*self.y
    def Div(self):
        return self.x/self.y
class s_calc(basic_calc):
    def ini__(self,x,y):
        self.x=x
        self.y=y
    def Factorial(self):
        return (math.factorial(self.x),math.factorial(self.y))
    def XpowerY(self):
        return pow(self.x,self.y)
    def Log(self):
        return(math.log(self.x),math.log(self.y))
    def ln(self):
        return(math.log(self.x),math.log(self.y))
num1=int(input("Enter First Number: "))
num2=int(input("Enter Second Number: "))
BC=basic_calc(num1,num2)
SC=s_calc(num1,num2)
print("Sum=",BC.Sum())
print("Sub=",BC.Sub())
print("Multiplication=",BC.Mul())
print("Division=",BC.Div())
print("Factorial=",SC.Factorial())
print("X power Y=",SC.XpowerY())
print("Log=",SC.Log())
print("Natural Log=",SC.ln())

```

**Result/Output:**



```
In [94]: runfile('C:/Users/Student/BasicCalculator.py', wdir='C:/Users/Student')
```

```
Enter First Number: 5
```

```
Enter Second Number: 10
```

```
Sum= 15
```

```
Sub= -5
```

```
Multiplication= 50
```

```
Division= 0.5
```

```
Factorial= (120, 3628800)
```

```
X power Y= 9765625
```

```
Log= (1.6094379124341003, 2.302585092994046)
```

```
Natural Log= (1.6094379124341003, 2.302585092994046)
```

```
In [95]: |
```

---

**Analysis/Conclusion:** functions inside a class should must have self argument

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:

### Basic Calculator

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
Created on Wed Mar 3 08:39:27 2021
```

```
@author: Student
```

```
"""
```

```
class basic_calc():
```

```
    def __init__(self,x,y):
```

```
        self.x=x
```

```
        self.y=y
```

```
    def Sum(self):
```

```
        return self.x+self.y
```

```
    def Sub(self):
```

```
        return self.x-self.y
```

```
    def Mul(self):
```

```
        return self.x*self.y
```

```
    def Div(self):
```

```
        return self.x/self.y
```

```
num1=int(input("Enter First Number: "))
```

```
num2=int(input("Enter Second Number: "))
```

```

BC=Basic_Calc(num1,num2)
print("Sum=",BC.Sum())
print("Sub=",BC.Sub())
print("Multiplication=",BC.Mul())
print("Division=",BC.Div())

```

## Scientific calculator

```

# -*- coding: utf-8 -*-
"""

```

Created on Wed Mar 3 10:37:22 2021

```

@author: Student
"""

```

```

import BasicCalculator
import math
class s_calc(BasicCalculator.basic_calc):
    def ini__(self,x,y):
        self.x=x
        self.y=y
    def Factorial(self):
        return (math.factorial(self.x),math.factorial(self.y))
    def XpowerY(self):
        return pow(self.x,self.y)
    def Log(self):
        return(math.log(self.x),math.log(self.y))
    def ln(self):
        return(math.log(self.x),math.log(self.y))
SC=s_calc(BasicCalculator.num1,BasicCalculator.num2)
print("Factorial=",SC.Factorial())
print("X power Y=",SC.XpowerY())
print("Log=",SC.Log())
print("Natural Log=",SC.ln())

```

## Result/Output:

```

In [92]: runfile('C:/Users/Student/ScientificCalculator.py', wdir='C:/
Users/Student')

```

Enter First Number: 5

Enter Second Number: 10

Sum= 15

Sub= -5

Multiplication= 50

Division= 0.5

Factorial= (120, 3628800)

X power Y= 9765625

Log= (1.6094379124341003, 2.302585092994046)

Natural Log= (1.6094379124341003, 2.302585092994046)

```

In [93]: |

```

**Analysis/Conclusion:** we can import other files to new file.