# LAB ACTIVITY 1:
# Introduction to Basic Operations in Python

## Learning Outcomes:

**By the end of this laboratory session, you should be able to:**

1. **Display the use of different types of literal.**
2. **Display the implementation of Python operators.**

**Hardware/Software: Computer, Phyton 3.5 or above.**

## Activity 1A(i)


Activity Outcome: Display the use of different types of literal in phyton.

Procedure:

**Step 1:** Open Code editor and type the following code:

```
x = str("Hello World")
x = int(20)
x = float(20.5)
x = 5
print(type(x))

#Legal variable names:
myvar = "John"
my_var = "John"
_my_var = "John"
myVar = "John"
MYVAR = "John"
myvar2 = "John"

#Illegal variable names:
2myvar = "John"
my-var = "John"
my var = "John"
```

**Step 2:** Save, compile and run the program. Save the program as `Act1A.py`. Write the output in the area below.

**Output:**

```
  File "C:\Users\P340\Documents\Python Code\Act1A(i).py", line 16
    2myvar="John"
    ^
SyntaxError: invalid decimal literal

***Repl Closed***
```

# Activity 1A(ii)


Activity Outcome: Display the use of different types of literal in phyton

Procedure:

**Step 1:** Open Code editor and type the following code:

```
#variables
prog = "DDT4"
noOfStudent = 80
print(prog)
print(noOfStudent)

#change type of variables
x = 4        # x is of type int
x = "Sally" # x is now of type str
print(x)

#assign values to multiple variables in one line
f1, f2, f3= "Orange", "Banana", "Cherry"
print(f1)
print(f2)
print(f3)

#assign the same value to multiple variables in one line
f1, f2, f3= "apple"
print(f1)
print(f2)
print(f3)
```

```
print(type(prog))
```

**Step 2:** Save, compile and run the program. Save the program as `Act1A.py`. Write the output in the area below.

**Output:**


```
DDT4
80
Sally
Orange
Banana
Cherry
Traceback (most recent call last):
  File "C:\Users\P340\Documents\Python Code\LabActivity1\Act1A(ii).py", line 19, in <module>
    f1,f2,f3="apple"
ValueError: too many values to unpack (expected 3)

***Repl Closed***
```

# Activity 1B


Activity Outcome: Arithmetic Operators

Procedure:

**Step 1:** Open code editor and type the following code:

```
a = 21
b = 10
c = 0
c = a + b
print ("Line 1 - Value of c is ", c)

c = a - b
print ("Line 2 - Value of c is ", c )

c = a * b
print ("Line 3 - Value of c is ", c)

c = a / b
print ("Line 4 - Value of c is ", c )

c = a % b
print ("Line 5 - Value of c is ", c)

a = 2
b = 3
c = a**b
print ("Line 6 - Value of c is ", c)

a = 10
b = 5
c = a//b
print ("Line 7 - Value of c is ", c)
```

**Step 2:** Save, compile and run the program. Save the program as `Act1B.py`. Write the output in the area below.

**Output:**

```
Line 1 - value of c is 31
Line 2 - value of c is 11
Line 3 - value of c is 210
Line 4 - value of c is 2.1
Line 5 - value of c is 1
Line 6 - value of c is 8
Line 7 - value of c is 2
```

# Activity 1C


Activity Outcome : Expressions and Operator Precedence.

Procedures:

**Step 1:** Open code editor and type the following code:

```
a = 20
b = 10
c = 15
d = 5

print ("a:%d b:%d c:%d d:%d" % (a,b,c,d ))
e = (a + b) * c /  d           #( 30 * 15 ) / 5
print ("Value of (a + b) * c / d is ",  e)

e = ((a + b) * c) / d          # (30 * 15 ) / 5
print ("Value of ((a + b) * c) / d is ",  e)

e = (a + b) * (c / d)     # (30) * (15/5)
print ("Value of (a + b) * (c / d) is ",  e)

e = a + (b * c) / d           #  20 + (150/5)
print ("Value of a + (b * c) / d is ",  e)
```

S

**Step 2:** Save, compile and run the program. Save the program as `Act1C.py`. Observe the output.

**Output:**

```
a:20 b:10 c:15 d:5
Value of ((a + b) * c / d is 90.0
Value of ((a + b) * c ) / d is 90.0
Value of ( a + b ) * ( c / d ) is 90.0
Value of a + (b * c) / d is 50.0
```

# Activity 1D


Activity Outcome : Assignment Operators

Procedures:
**Step 1:** Open code editor and type the following code:

```
a = 21
b = 10
c = 0

c = a + b
print ("Line 1 - Value of c is ", c)

c += a
print ("Line 2 - Value of c is ", c )

c *= a
print ("Line 3 - Value of c is ", c )
```

**Step 2:** Save, compile and run the program. Save the program as `Act1D.py`. Observe the output.

**Output:**

```
Line 1 - Value of c is 31
Line 2 - Value of c is 52
Line 3 - Value of c is 1092
```