Institute of Information Technology

Jahangirnagar University

Savar, Dhaka-1342

Lab Manual

Course Code: ICT-4202

Course Title: Digital Image Processing Lab

Lab No.: 1

Lab Title: INTRODUCTION TO PYTHON

Prepared by

Mehrin Anannya

Assistant Professor

Institute of Information Technology

Jahangirnagar University

Lab Title: INTRODUCTION TO PYTHON

OBJECTIVE:

To introduce students to the python programming language.

Lab Contents:

1. Comment
2. print()
3. Operators
    i. + plus
    ii. - minus
    iii. * multiply
    iv. ** power
    v. / divide
    vi. // divide and floor
    vii. % modulus
4. Variables
5. Strings
6. LISTS
7. Indentation
8. Boolean operations
9. Branching
10. Looping
11. Functions

Theory with Hands on Practice:

1. Comment

    # is used for commenting.

    Example: # This is a comment.

2. Whatever you put inside parentheses of print function will be your output

   Example : print (5); print (10); print (12+6);

3. Operators

```
print(5+6)
print(5*6)
print(2**4)
print(4/3)
print(7//3)
print(4%2)
```

4. Variables

```
x=4
print(type(x))
print(4)
x,y,z=3.1417, 67,5
print(x+y)
print(type(x))
z=True #First letter should be zero in declaring v
print(z)
print(type(z))
t=str(56)
print(t)
print(type(t))


x=8
y=9
print(x,"-",y, ":", x-y)
```

5. Strings

```
a='py'
b='charm'
print (a + b)      # + will join the two strings
print (a,b)
print (a*5)
#print (a + 5)      # Type Error: must be str, not int
print (a + str (5))
t = """Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua."""
print(t)
```

6. Lists

```
x = [1,2,3,4]
```

note: list are not arrays because arrays are of single data type however list may contain variables of different data types.

```
x=int(input ("Enter a number="))  # input ( ) function always input string,
                                  # int ( ) function is used to convert string to int
print(x)
print(type(x))


y, z = 3.14, "HELLO"
li = [x, y, z, 4]
print(li)
```

List indexes

```
# From left to right:  0 -> 1 ->  2
# From right to left:  -1 -> -2 -> -3
```

```
print (li [2] [-4])
```

slicing #list [start: end: optional step size] ; start is inclusive and end is exclusive

LISTS

```
x = [1,2,3,4]
```

note: list are not arrays because arrays are of single data type however list may contain variables of different data types.

```
x=int (input ("Enter a number=")) # input ( ) function always input string,
    # int ( ) function is used to convert string to int
        print (li [1:3])
        print (li [0:4:2])
        print (li [:])
        print (li [0:])
        print (li [:3])
        print (li [2] [1:4])
        #Deleting something from a list del (li [2])
        del(li[3])
        print(li)
```

copy

```
w = [1, 2, 3, 4]
x = w                  # The assignment copies the reference to the original list
y = w [:]              # Slicing creates a new list
x [0] = 6
y [1] = 9
print(w)
```

7. Indentation

Whitespace is important in Python. Actually, whitespace at the beginning of the line is important. This is called indentation. Leading whitespace (spaces and tabs)

at the beginning of the logical line is used to determine the indentation level of the logical line, which in turn is used to determine the grouping of statements.

Example:

```
x=2
if x==10:
        print ("this is inside if block")
        print ("this is also inside if block")
print("this will print always")
```

8. Boolean operations

< (less than)
>(greater than)
<= (less than or equal to)
>= (greater than or equal to)
== (equal to)
! = (not equal to)
not (boolean NOT) if not x:
and (boolean AND) if x==2 and y>4:
or (boolean OR) if x==2 or y>4:

9. The IF statement

```
number = 23
if number == 24:
    print ('number is equal to 24')
elif number<24:
    print ("Number is less than 24")
else:
    print ("Number is higher than 24")
```

10. The while statements

```
number = 23
while number<50:
        print(number)
        number+=1
else:
        print ("wow there is an else for while statement as well")
```

11. The for statement

```
    #Indentation
    for i in range (10):
        if i==6:
            break
        print(i)
        for i in range (3, 10):
            print(i)
            if i==6:
            continue
        for i in range (1, 10, 3):
            print(i)
```

12. Functions

```
        def print_max (a, b=0):
            if a > b:
                print (a, 'is maximum')
            elif a == b:
                print (a, 'is equal to', b)
            else:
```

```
            print (b, 'is maximum')
    print_max (3, 4)
    x = 5
    y = 7
    print_max (x, y)
```

Keyword arguments

```
    def func (a, b=5, c=10):
        print ('a is', a, 'and b is', b, 'and c is', c)
    func (3, 7)
    func (25, c=24)
    func (c=50, a=100)
```

Return

```
    def max (x, y):
        if x > y:
            return x
        else:
            return y
    m=max (3,5)
    print(m)
```

Tasks:

Task – 1:

x = [[1, 2, 3, 4, 5], [21, 22, 23, 24, 25], [31, 32, 33, 34, 35]

a. Write python code using python indexing and slicing for the following output. Use only one print statement for each:
   i. [31, 32, 33, 34, 35]
   ii. 23
   iii. [22,23]
   iv. [1, 3, 5]

b. Declare y = [0, 0, 0], now using for loop write average of first list in list ‘x’ on first index of list y and so on. The print(y) should give the output: [3.0, 23.0, 33.0]

Task – 2:

x = [1, 3, 5, 6, 7, 8, 6, 1, 2, 3]

y = [0, 0, 0, 0, 0, 0, 0, 0]

a. Write python code using a while loop that writes the average of the first three items on the first index of y and so on. The print(y) should give the following output

   Output : [3.0, 4.666666666666667, 6.0, 7.0, 7.0, 5.0, 3.0, 2.0]

b. Define a function that takes the list as an argument and returns the average. Then calculate the average of x and y.