

IPP Lab Assignment-3

1. What will be the output produced by each of the following function calls:
 - a. `math.ceil(65.65)`
 - b. `math.ceil(65.47)`
 - c. `math.fabs(-67.58)`
 - d. `math.fabs(3)`
 - e. `math.exp(2.7)`
 - f. `math.log(45,2)`
 - g. `math.log10(1000)`
 - h. `math.pow(4, 1/2)`
 - i. `math.sqrt(121)`
 - j. `math.radians(30)`
 - k. `math.degrees(math.pi/2)`
2. Give the range in which value of variable x may lie on execution of the following statements: `import random`

`x = random.random() + 5`

3. Evaluate the following expressions using Python shell. Assume that ASCII coding scheme is used for character data.
 - a. `abs(-5.4)`
 - b. `abs(15)`
 - c. `chr(72)`
 - d. `round(-24.9)`
 - e. `float(57)`
 - f. `complex('1+2j')`
 - g. `divmod(5,2)`
 - h. `float(57)`
 - i. `pow(9,2)`
 - j. `max(97, 88, 60)`
 - k. `min(55, 29, 99)`
 - l. `max('a', 'b', 'AB')`

4. Consider the following function:

```
def nMultiple(a = 0, num = 1):  
    return a*num
```

What will be the output produced when the following calls are made:

- a. `nMultiple(5)`
 - b. `nMultiple(5,6)`
 - c. `nMultiple(num = 7)`
 - d. `nMultiple(num = 6, a = 5)`
 - e. `nMultiple(5, num = 6)`
5. Study the program segment given below. Give the output produced, if any.

```
def func():  
    pass  
a=func()  
print(a)
```

6. Study the program segments given below. Give the output produced, if any.

a.)

```
def say (message, times=2):  
    print(message*times)
```

```
say ('Hello')  
say('World',5)
```

b.)

```
def fun(a=2,b=3,c=7):  
    d= a+b+c  
    print(d)  
    print(fun (2))
```

7. Define a function to find the sum of even digits of a four-digit number without using control structures.)
8. Using a function evaluate the value of the arithmetic expression taken from the user. Hint: Expression will act as an argument while defining function.
9. What does a function return by default in Python? Define a function that does not return any value, store the function call in a variable and check the value of that variable.
10. Write a function named as 'UpperCase' which converts the lower case alphabet to uppercase alphabet. Also, assert that the entered alphabet by user is valid lowercase alphabet(use ord()). Write a function main that accepts inputs from the user interactively and converts the lowercase alphabet to uppercase using the function 'UpperCase'(chr() is used to convert ascii value to its corresponding character and ord() converts returns the ascii value of the given character)

```
>>> chr(97)  
'a'  
>>> ord('a')  
97
```

11. Write a Python function to swap the values of two variables.
12. Write a function `areaTriangle` that takes the lengths of three sides: `side1`, `side2`, and `side3` of the triangle as the input parameters and returns the area of the triangle as the output. Also, assert that sum of the length of any two sides is greater than the third side. Write a function `main` that accepts inputs from the user interactively and computes the area of the triangle using the function `areaTriangle`.
13. Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument.
14. Write a Python function to check whether a number falls within a given range(range has to be entered as command line arguments).
15. Observe carefully the below function

```
def fun(a=0, b=1):  
    return (a**2 + b**2)
```

What will be the output for each call made below?

 - a.) `fun(2,a=3)`
 - b.) `fun(b=3,2)`
 - c.) `fun(3,b=2)`
 - d.) `fun(a=4,5)`

16. What will be the output of following code?

```
x = -5  
def display(x):  
    print(x)  
    x = 5  
    print(x)  
display(x)  
print(x)
```

17. Create the following scripts `importedModule` and `mainModule` in the working directory, execute the script `mainModule` and justify the output

- **importedModule.py**
def test1():
 print('test1 in imported module')
def test2():
 print('test2 in imported module')
test1()
test2()
- **mainModule.py**
import importedModule
print('hello')

18. Create the following scripts f1 and f2 in the working directory, execute the script f2 and justify the output.

- **f1.py**
def display():
 print("hello")

display()
print("ITER")

- **f2.py**
import f1
f1.display()

19. Create the following scripts f1 and f2 in the working directory, execute the script f2 and justify the output.

- **f1.py**
def display():
 print("hello")

if __name__ == '__main__':
 display()
 print("ITER")

- **f2.py**
import f1
f1.display()