Experiment 1

Objective : To show uses of the operators in Python

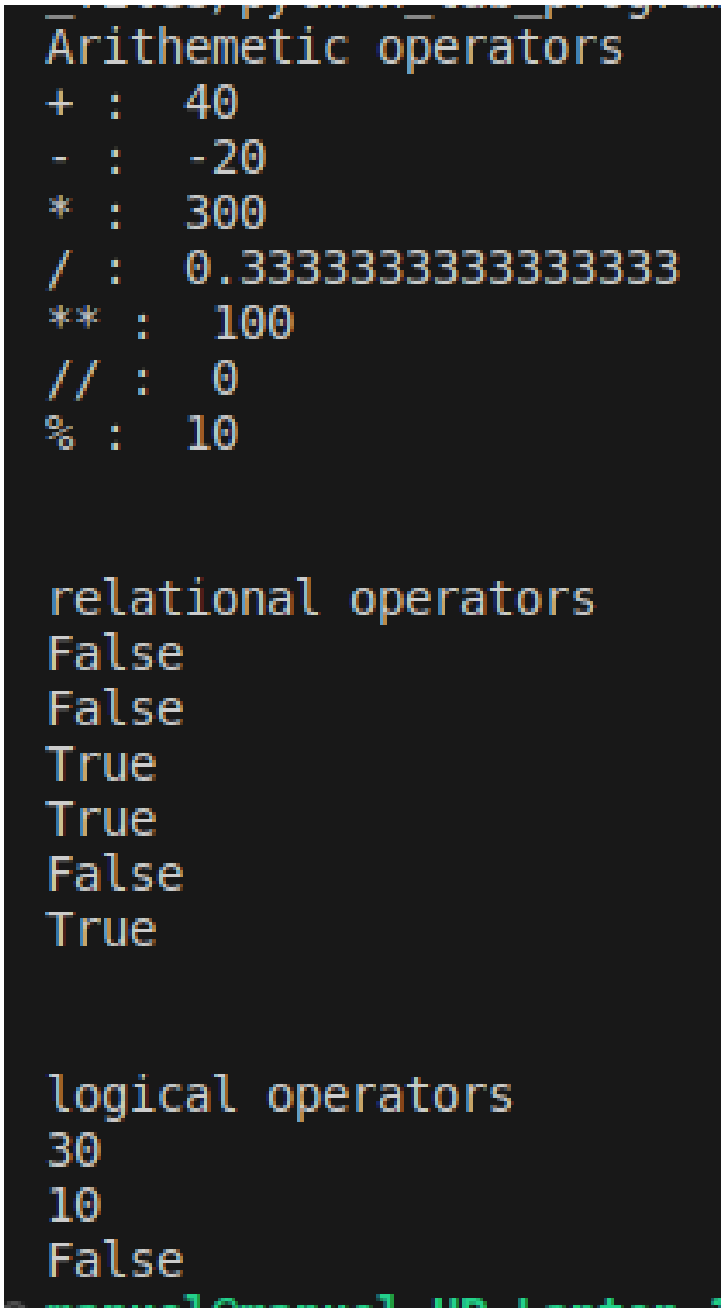
Code :

```
#program 1
a=10
b=30
#arithemetic operators
print("Arithemetic operators")
print("+ : ",a+b)
print("- : ",a-b)
print("* : ", a*b)
print("/ : ", a/b)
print("** : ", a**2)
print("// : ", a//b)
print("% : ", a%b)
print("\n")

#relational operators
print("relational operators")
print(a>b)
print(a>=b)
print(a<=b)
print(a<b)
print(a==b)
print(a!=b)
print("\n")

#logical operators
print("logical operators")
print(a and b)
print(a or b)
print(not a)
```

Output :


Arithemetic operators
+ : 40
- : -20
* : 300
/ : 0.3333333333333333
** : 100
// : 0
% : 10
relational operators
False
False
True
True
False
True
logical operators
30
10
False

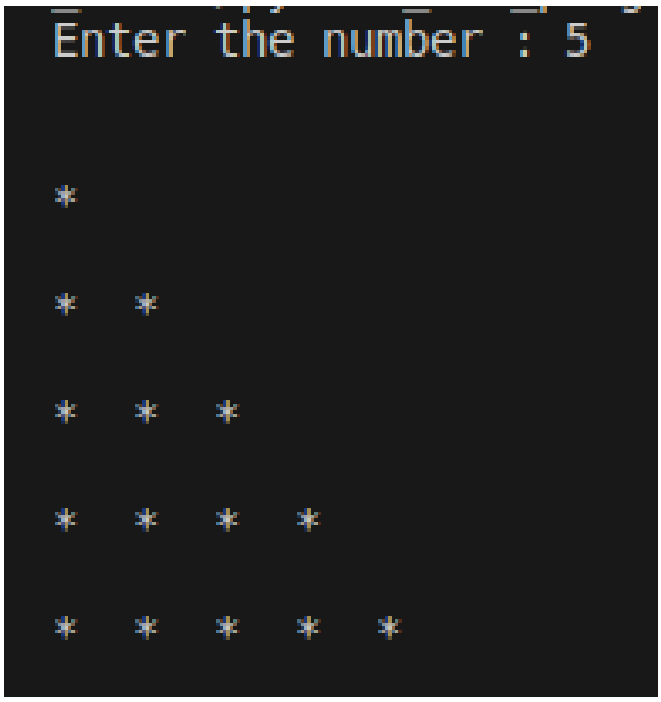
Experiment 2

Objective : To display the star pattern using for loops

Code :

```
num=int(input("Enter the number : "))
for i in range(num+1):
    for j in range(i):
        print("* ", end=" ")
    print("\n")
```

Output :



Experiment 3

Objective : To perform string operations using inbuilt methods

Code :

```
#program 3

a="Hello World"
print("String operations\n")
print("Original String : ",a)
print("Converts all to lowercase : ", a.lower())
print("Converts all to uppwercase : ",a.upper())
print("Swaps the cases of alphabets : ",a.swapcase())
print("Checks whether the string is alphabet or not : ", a.isalpha())
print("Checks whether the string is digit or not : ",a.isdigit())
print("Shows the first index of the letter in given string : ",a.index("r"))
print("counts the letter it occoured : ",a.count("o"))
print("splits the string using delimeter : ",a.split(" "))
print("Replaces the word with the given string : ",a.replace("Hello", "Hi"))
print("Checks whether the string ends with given word : ",a.endswith("d"))
```

Output :

```
String operations

Original String :  Hello World
Converts all to lowercase :  hello world
Converts all to uppwercase :  HELLO WORLD
Swaps the cases of alphabets :  hELLO wORLD
Checks whether the string is alphabet or not :  False
Checks whether the string is digit or not :  False
Shows the first index of the letter in given string :  8
counts the letter it occoured :  2
splits the string using delimeter :  ['Hello', 'World']
Replaces the word with the given string :  Hi World
Checks whether the string ends with given word :  True
```

Experiment 4

Objective : To perform list operations using inbuilt methods

Code :

```
a=[5,3,6,7,4,1,9,2]
print("Original list : ",a)
print("Sorts the list in ascending order : ",a.sort())
print("returns the index of the given element : ", a.index(5))
print("Counts the element it occoured : ",a.count(3))
print("removes the element using the index value : ", a.pop())
print("Removes the element using the given value : ", a.remove(5))
print(a)
print("Adds the value to the list : ",a.append(5))
print(a)
print("Reverses the list : ",a.reverse())
print(a)
print("Adds the element to a specific index : ",a.insert(5,10))
print(a)
print("",a.extend([69,89,79]))
print(a)
```

Output :

```
_files/python_lab_programs/new_lab_programs/4.py
Original list :  [5, 3, 6, 7, 4, 1, 9, 2]
Sorts the list in ascending order :  None
returns the index of the given element :  4
Counts the element it occoured :  1
removes the element using the index value :  9
Removes the element using the given value :  None
[1, 2, 3, 4, 6, 7]
Adds the value to the list :  None
[1, 2, 3, 4, 6, 7, 5]
Reverses the list :  None
[5, 7, 6, 4, 3, 2, 1]
Adds the element to a specific index :  None
[5, 7, 6, 4, 3, 10, 2, 1]
 None
[5, 7, 6, 4, 3, 10, 2, 1, 69, 89, 79]
manuel@manuel-HP-Laptop-15-da0xxx:~/Downloads/Anirud
```

Experiment 5

Objective : To implement the stack and queue using a list.

Code :

```
#program 5

class queue_implemntation:
    def __init__(self,list1=[]):
        self.list1=list1

    def enqueue(self,n):
        self.list1.append(n)
    def display(self):
        if(len(self.list1)==0):
            print("empty queue")
        else:
            for i in range(len(self.list1)-1,-1,-1):
                print(self.list1[i],end=" ")
            print("\n")

    def dequeue(self):
        if(len(self.list1)==0):
            print("empty queue")
        else:
            self.list1.pop(0)


class stack_implementation:
    def __init__(self,list1=[]):
        self.list1=list1
    def push(self,n):
        self.list1.append(n)
    def display_stack(self):
        for i in range(len(self.list1)-1,-1,-1):
            print(self.list1[i],end=" ")
        print("\n")
    def pop_stack(self):
        if(len(self.list1)==0):
            print("Stack is empty")
        else:
            self.list1.pop()
```

```
obj1=stack_implementation()
obj1.push(1)
obj1.display_stack()
obj1.push(2)
obj1.push(3)
obj1.display_stack()
obj1.pop_stack()
obj1.display_stack()


obj2=queue_implemntation()
obj2.enqueue(3)
obj2.enqueue(4)
obj2.enqueue(1)
obj2.enqueue(69)
obj2.display()
obj2.dequeue()
obj2.display()
```

Output :

```
1

3 2 1

2 1

69 1 4 3

69 1 4
```

Experiment 6

Objective : To use dictionary and make a dictionary of faculty and students and store them seprately in lists.

Code :

```
dict1=dict()

def addiemsindict(a,b,c):
    dict1[a]={"Faculty":b,"Student":c}

while 1==1:
    deptname=input("Enter the name of department : ")
    fname=input("Enter the name of the faculty : ")
    sname=input("Enter the name of the student : ")
    addiemsindict(deptname,fname,sname)
    choice=int(input("are you done(1 for yes , 0 for no) : "))
    if(choice==1):
        break
    else:
        pass
print(dict1)

student_list=[]
faculty_list=[]

for i in dict1.keys():
  student_list.append(dict1[i]["Student"])
  faculty_list.append(dict1[i]["Faculty"])

print(student_list)
print(faculty_list)
```

Output :

```
_files/python_lab_programs/new_lab_programs/6.py
Enter the name of department : SOICT
Enter the name of the faculty : A
Enter the name of the student : B
are you done(1 for yes , 0 for no) : 0
Enter the name of department : SOBT
Enter the name of the faculty : C
Enter the name of the student : D
are you done(1 for yes , 0 for no) : 1
{'SOICT': {'Faculty': 'A', 'Student': 'B'}, 'SOBT': {'Faculty': 'C', 'Student': 'D'}}
Student List :  ['B', 'D']
Faculty List :  ['A', 'C']
```

Experiment 7

Objective : To show the use of lambda expression.

Code :

```
#program 7

A=lambda x:x+6
print(A(6))

list1=list(map(int,input().split()))
print(list1)

y=lambda x,z: z if A(z)+3>x else 6

print(y(6,3))
```

Output :

```
_files/python_lab_programs/new_
12
1 2 3 4 5 6 7 8 9 0
[1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
3
```

Experiment 8

Objective : To demonstrate the use of File writing and reading in text file

Code :

```python
#program 8

with open("file1.txt", "w") as f:
    while(1==1):
        line=input("enter the lines : ")
        f.write(line)
        f.write("\n")
        choice=input("are you done(Y/N) : ")
        if(choice.lower()=="y"):
            break
        else:
            pass
    f.close()
    print("Written in file Successfully")

with open("file1.txt", "r") as g:
    print("Reading the file\n")
    print(g.read())
```

Output :

```
enter the lines : Hi I learn Python
are you done(Y/N) : n
enter the lines : I also learn Java
are you done(Y/N) : y
Written in file Successfully
Reading the file

Hi I learn Python
I also learn Java
```

Experiment 9

Objective : To demonstrate error handling

Code :

```
#program 9

try:
    a=int(input("Enter the number : "))
    print(a/2)
    print(a/0)
except (ArithmeticError, ValueError):
    print("An error Occoured\n")
```

Output :

```
_files/python_lab_programs/new
Enter the number : 4
2.0
An error Occoured
```

```
_files/python_lab_programs/new
Enter the number : a
An error Occoured

manuel@manuel-HP-Laptop-15-da
```

Experiment 10

Objective : To demonstrate Multiple inheritance using classes

Code :

```python
#program 10

class Father:
    def fatherwork(self):
        print("I work as a software engineer")

class Mother:
    def motherwork(self):
        print("I work as a Data Analyst")

class Child(Father, Mother):
    pass

c1=Child()
c1.fatherwork()
c1.motherwork()
```

Output :

```
_files/python_lab_programs/new_
I work as a software engineer
I work as a Data Analyst
```

Experiment 11

Objective : To use Numpy and pandas to generate a list.

Code :

```
#program 11

import pandas as pd
import numpy as np

# Creating empty series
ser = pd.Series()
print("Pandas Series: ", ser)

# simple array
data = np.array(['g', 'e', 'e', 'k', 's'])

ser = pd.Series(data)
print("Pandas Series:\n", ser)
```

Output :

```
Pandas Series:  Series([], dtype: object)
Pandas Series:
 0    g
1    e
2    e
3    k
4    s
dtype: object
```