

Practical

(a) Write a python script to demonstrate Operators.

```
#Arithmetic operator
a=5
b=4
#addition
c=a+b
print(a, "+", b, "=", c)
#subtraction
c=a-b
print(a, "-", b, "=", c)
#multiplication
c=a*b
print(a, "*", b, "=", c)
#division(float)
c=a/b
print(a, "/", b, "=", c)
#division(floor)
c=a//b
print(a, "//", b, "=", c)
#modulus
c=a%b
print(a, "%", b, "=", c)
#power
c=a**b
print(a, "**", b, "=", c)
```

```
=====
5 + 4 = 9
5 - 4 = 1
5 * 4 = 20
5 / 4 = 1.25
5 // 4 = 1
5 % 4 = 1
5 ** 4 = 625
>>> |
```

```
#Relational operator
a=9
b=3
#>
c=a>b
print(a, ">", b, "=", c)
#<
c=a<b
print(a, "<", b, "=", c)
#==
c=a==b
print(a, "==", b, "=", c)
#!=
c=a!=b
print(a, "!=", b, "=", c)
#>=
c=a>b
print(a, ">=", b, "=", c)
#<=
c=a>b
print(a, "<=", b, "=", c)
```

```
9 > 3 = True
9 < 3 = False
9 == 3 = False
9 != 3 = True
9 >= 3 = True
9 <= 3 = True
>>> |
```

```
#Assignment operator
a=9
b=3
d=46
# =
c=a+b
print(a, "+", b, "=", c)
# +=
d+=a
print(d)
# -=
d-=a
print(d)
# *=
d*=a
print(d)
# /=
d/=a
print(d)
# %=
d%=a
print(d)
# **=
d**=a
print(d)
# //==
d//=a
print(d)
```

```
=====
9 + 3 = 12
55
46
414
46.0
1.0
1.0
0.0
>>> |
```


```
#logical
x = 10
y = 0
# Output: x and y is False
print('x and y is',x and y)
x = 10
y = 0
# Output: x and y is True
print('x or y is',x or y)
x = 10
y = 0
# Output: not x is false
print('not x is',not x )
```

```
x and y is 0
x or y is 10
not x is False
>>>
```

```
#bitwise
x = 1
y = 8
# AND
a=x&y
print(a, "=", x, "&", y)
# |
a=x|y
print(a, "=", x, "|", y)
# ~
a=~x
print(a, "= ~", x)
# ^
a=x^y
print(a, "=", x, "^", y)
# >>
a=x>>y
print(a, "=", x, ">>", y)
# <<
a=x<<y
print(a, "=", x, "<<", y)
```

```
=====
0 = 1 & 8
9 = 1 | 8
-2 = ~ 1
9 = 1 ^ 8
0 = 1 >> 8
256 = 1 << 8
>>> |
```


(b) Write a python script to demonstrate Type casting (Implicit and Explicit)

 script.py - D:/21se02ml006/script.py (3.9.7)
File Edit Format Run Options Window Help

```
number1=1
print(type(number1))
number2=2.0
print(type(number2))
ans=number1+number2
print(number1, "+", number2, "=", ans)
print(type(ans))
```



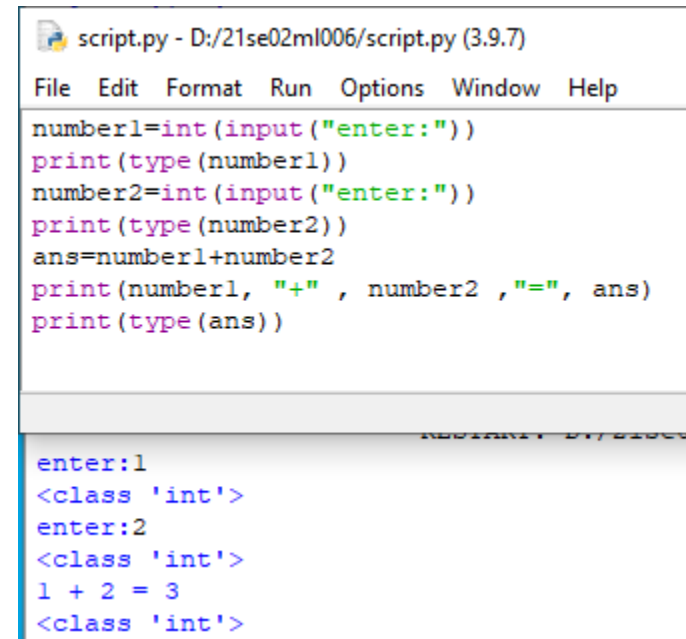
```
<class 'int'>
<class 'float'>
1 + 2.0 = 3.0
<class 'float'>
```

 script.py - D:/21se02ml006/script.py (3.9.7)
File Edit Format Run Options Window Help

```
number1=input("enter:")
print(type(number1))
number2=input("enter:")
print(type(number2))
ans=number1+number2
print(number1, "+", number2, "=", ans)
print(type(ans))
```



```
enter:1
<class 'str'>
enter:2
<class 'str'>
1 + 2 = 12
<class 'str'>
```



The image shows a screenshot of a Python script editor window. The title bar reads "script.py - D:/21se02ml006/script.py (3.9.7)". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The script code is as follows:

```
number1=int(input("enter:"))  
print(type(number1))  
number2=int(input("enter:"))  
print(type(number2))  
ans=number1+number2  
print(number1, "+", number2, "=", ans)  
print(type(ans))
```

Below the script, the output of the program is displayed:

```
enter:1  
<class 'int'>  
enter:2  
<class 'int'>  
1 + 2 = 3  
<class 'int'>
```