21/7/25

Find outputs (HOME WORK)

```
a={
  print('Hyd'),
  print('Sec'),
  print('cyb').
  }
print(type(a)) # <class 'set'>
Print(a)       # {None}
Print(len(a))  # 1
```

Anonymous object find outputs

```
a=25
print(id(a)) # 140712660715176
a=35
print(id(a)) # 140712660715496
```

Find outputs (Home work)

```
a=25.7
print(id(a)) # 2142696485296
Print(a)     # 25.7
a=35.6       #
print(id(a)) # 2142705550544
print(a)     # 35.6
b=True
print(id(b)) # 140712659829168
b=False
print(id(b)) # 140712659829200 Reference
```

```
c = None      Address of obj None
print(id(c))  # 140712659829232
c = None
print(id(c))  # 140730296013 28
# Anonymous Object demo program
_ = 25
print(_)              # prints 25
print(type(_))        # prints <class 'int'>
a,_,c = 10,20,30
print(a)              # prints 10
print(_)              # prints 20 (From tuple is unpacking)
print(c)              # prints 30
for _ in range(5):
    print(_, 'Hello') # prints numbers 0 to 4 with
                      'Hello'.
# Find outputs (Homework)
a = 'Hyd'
print(id(a)) # prints id of 'Hyd'
a[1] = 'e'   # Error! strings are immutable
a = 'Sec'
print(id(a)) # prints id of 'Sec'
b = (10,20,15,18)
print(id(b))  # prints id of the original tuple
b[2] = 19     # Error! Tuples are also immutable
b = (30,40,35,32)
print(id(b))  # New id, since b now refers to a new tu
c = range(5)
print(id(c))  # prints id of range(0,5)
c[3] = 10 # Error range obj are immu
c = range(5)
print(id(c)) #
```

Find outputs (Home work)

```
a=25
b=25
print(a is b)  # True
c='Hyd'
d='Hyd'
print(c is d)  # True
e=False
f=False
print(e is f)  # True
g=range(10)
h=range(10)
print(g is h)  # False
```

Find outputs (Home work)

```
a=[10,20,15,18]
b=[10,20,15,18]
print(a is b)        # False
c={10,20,30,40}
d={10,20,30,40}
print(c is d)        # False
e=(10,20,15,18)
f=(10,20,15,18)
print(e is f)        # ~~False~~ True
g={10,20,15,18}
h={10,20,15,18}
print(g is h)        # False
```

Find outputs (Home work)

```
print(10+20)          # 30
print(10.8+20.6)      # 31.4
print(3+4j+5+6j)      # (8+10j)
print(True+False)     # 1
print('Hyder'+'abad')  # Hydderabad
print('sankar'+'Dayal'+'sarma')  # sankarDayalSarma
print('10'+'20')      # 1020
print([10,20,30]+[1,2,3])  # [10,20,30,1,2,3]
print((25,10.8,'Hyd')+(3+4j,True,None))
# (25,10.8,'Hyd',(3+4j),True,None)
print((25,10.8,'Hyd')+(3+4j,True,
print({10,20}+{30,40})  # Error set not concatenate
print({10:'Hyd'}+{20:'Sec'})  # Error
print(range(4)+range(5))  # Error
print(10+'20')  # Error
print([10,20,30]+5)  # Error
print([10,20,30]+[40,50,60])  # Error
```

Find outputs (Home work)

```
print(25*3)          # 75
print(10.8*25.6)     # 276.48
print(True*False)    # 0
```

```
print((3+4j)*(5+6j))   #(-9+38j)
print(3+4j*5+6j)       #(3+26j)
print('25'*3)          # 252525
print(3*'25')          # 252525
print('Hyd'*4)         # HydHydHydHyd
print([10,20,15]*2)    #[10,20,15,10,20,15]
print({10,20,15}*3.0)  # Error
print([25,10.8,'Hyd',True]*3)
#(25,10.8,'Hyd',True,25,10.8,'Hyd',True,25,10.8,'Hyd',True)
print({10,20,15}*2)    # Error
print({10:20,30:40}*2) # Error
print([10]*[20])       # Error
```

#Operator (/) demo program

```
print(9.0/2)   # 4.5
print(9/2.0)   # 4.5
print(9.0/2.0) # 4.5
print(9/2)     #4.5
print(10.5/2)  # 5.25
print(10/3)    # 3.333
print(10//2)   # 5
```

// operator demo program

```
print (9//2)  # floor integer of (4.5) = 4
print (9.0//2)  # floor integer of (4.0) = 4
print (9//2.0)  # floor integer of (4.0) = 4
print (9.0//2.0)  # floor integer of (4.5) = 4
print (10.5//2)  # floor integer of (5.25) = 5.0
print (10//3)  # floor integer of (3.33) = 3
print (10.0//3)  # floor integer of 3.33 = 3.0
print (8.5//3)  # floor integer of 2.83 = 2.0
print (18//4)  # floor integer of 4.5 = 4
print (-18//4)  # floor integer of -4.5 = -5
print(-(18//4))  # floor integer of 4 = -4
```

%operator demo program

```
print (9%5)    # 4
print (9.0%5)  # 4.0
print (9%5.0)  # 4.0
print (9.0%5.0) # 4.0
print (10.5%2)  # 0.5
print (8.9%3)  # 2.9
```

Find outputs

print (7/0) # Error

print (7//0) # Error

print (7%0) # Error

** operator demo program

print (3**4) # 81

print (10**-2) # 0.01

print (4**3**2) # Right to left associativity = 262144

print (3+4*5-32/2**3) # 19.0

Relational operators demo program (Homework)

print (9>=5) # True ; > is satisfied

print (9>=9) # True ; = is satisfied

print (9>=12) # False ; Both are not satisfied.

print (6<=8) # True ; < is satisfied.

print (6<=6) # True ; = is satisfied.

print (6<=4) # False ; Both are not satisfied.

print (9!=7) # True ; ! is satisfied

print (6==8) # False ; Both are not satisfied.

print (True>False) # True ; 1>0

print (3+4j == 3+4j) # True ; Both are complex numbers

print (3+4j == 5+6j) # False Both complex no. are not equal

print (3+4j != 5+6j) # True complex no. are not equal

```
print(10 == 10.0)  # True
print(3+4j > 3+4j)  # Error
```

Find outputs (Home work)

```
print('Rama' > 'Rajesh')  # True ; 'm' > 'j'
print('Rama' < 'sita')  # True 'R' < 's'
print('Hyd' == 'Hyd')  # True Match same strings
print('Rama' <= 'Ramana')  # True strings are considered smaller.
print('Rama Rao' >= 'Rama')  # True
print('Hyd' != 'sec')  # True.
print('HYD' < 'hyd')  # True uppercase letters come before lowercase letters.
```

~~print(t~~

chaining relational operators (Home work)

```
print(10 < 20 < 30)  # True
print(10 >= 20 < 30)  # False 10 is not >= 20
print(10 < 20 > 30)  # False.
print(1 < 2 < 3 < 4)  # True
print(1 < 2 > 3 > 1)  # False
print(4 > 3 == 3 > 2)  # True.
```

or operator demo program

```
print (True or False) # True
print (False or True) # True
print (True or True) # True
print (False or False) # False
print (10 or 20) # 10
print (0 or 20) # 20
print (-25 or 0) # -25
print ("" or 35) # 35
print (6j or "Hyd") # 6j
print (0.0 or 3+4j) # 3+4j.
print ('Hyd' or 10.8) # 'Hyd'
```

Not operator demo program

```
print (not True) # False
print (not False) # True
print (not 25) # False
print (not 0) # True
print (not 'Hyd') # False
print (not '') # True
print (not -10) # False
print (not not ('Hyd')) # True
```

Find outputs (Home work)

```
i = 10
i = not i > 14
print(i)    # True
print(not(6<4 or 9>=5 and 6!=6))  # True.
```

6<4 → False.

9>=5 → True

6!=6 → False

9>=5 and 6!=6 → True and False.

6<4 or (False) → False.

not(False) → True