3. Bitwise OR Operator (|)

```
In [6]:  s1={10,20,30}
         s2={20,15,25}
         s3=(s1.union(s2))
         print(s3,type(s3))

        {20, 25, 10, 30, 15} <class 'set'>
```

```
In [5]:  10|15

Out[5]:  15
```

```
In [8]:  s1={10,20,30}
         s2={20,15,25}
         s3=(s1|s2)
         print(s3,type(s3))

        {20, 25, 10, 30, 15} <class 'set'>
```

```
In [12]:  s1={1.2,2.3,3.5}
          s2={1.2,3.5,5.6}
          s3=s1|s2
```

```
In [14]:  print(s3,type(s3))

        {1.2, 2.3, 3.5, 5.6} <class 'set'>
```

```
In [2]:  s1={"Mahaboob","MRIIRS","CDOE"}
         s2={"Khan","TAQA","MRIIRS"}
         s3=s1|s2
         print(s3,type(s3))

        {'TAQA', 'CDOE', 'MRIIRS', 'Mahaboob', 'Khan'} <class 'set'>
```

```
In [4]:  {1,2,3}|{3,4,5}

Out[4]:  {1, 2, 3, 4, 5}
```

```
In [6]:  [10,20,30] | [30,40,50]

        ---------------------------------------------------------------------------
        TypeError                                 Traceback (most recent call last)
        Cell In[6], line 1
        ----> 1 [10,20,30] | [30,40,50]

        TypeError: unsupported operand type(s) for |: 'list' and 'list'
```

```
In [8]:  [10,20,30] or [30,40,50] # Use set() | set() if you want a union without duplicates

Out[8]:  [10, 20, 30]
```

```
In [4]: [] or [10]
```

```
Out[4]: [10]
```

```
In [8]: set("MRIIRS") | set("MRIIRS")
```

```
Out[8]: {'I', 'M', 'R', 'S'}
```

4.Bitwise AND Operator (&)

```
In [ ]: """The bitwise AND ( & ) operator compares each bit of two numbers:
        if both bit are 1, result is 1
        other wise, result is 0
        so it works on the binary representation of numbers."""
```

```
In [11]: 10 & 15
         # binary of 10 = 1100
         # binary of 15 = 1111
               # total  =1010 (which is 10 in decimal)
```

```
Out[11]: 10
```

```
In [5]: 3 & 4
```

```
Out[5]: 0
```

```
In [7]: 4 & 4
```

```
Out[7]: 4
```

```
In [9]: 100 & 100
```

```
Out[9]: 100
```

```
In [21]: "MRIIRS" & "KHAN"
         #In Python, the & operator is a bitwise operator, not a string operator.
         #It only works with integers (whole numbers) because it compares their binary bits.
         #Strings like "MRIIRS" and "KHAN" don’t have a direct binary representation for bit
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[21], line 1
----> 1 "MRIIRS" & "KHAN"

TypeError: unsupported operand type(s) for &: 'str' and 'str'
```

```
In [15]: 7&7|3
```

```
Out[15]: 7
```

In [27]:

```
s1={10,20,30}
s2={10,20,40}
s3=(s1 & s2)
print(s3,type(s3))
#In Python, the & operator on sets means intersection.The intersection of two sets
```

```
{10, 20} <class 'set'>
```

4.Bitwise complement operator(~)

In [32]:

```
#The symbol ~ is called Tilde
```

In [34]:

```
a=10 # ~a=-(a+1)
~a
```

Out[34]:

```
-11
```

In [36]:

```
a=100
~a
```

Out[36]:

```
-101
```

In [38]:

```
a=-102
~a
#~a=-(-102+1)
# -(-101)
# 101
```

Out[38]:

```
101
```

In [ ]:

```
""" THE CLASS WILL CONTINUE IN THE NEXT SESSION"""
```