Name: CH.SAI YASWANTH
Date:23/08/2025

# add() method demo program (Home work)

```
a = set()
a . add(True)
a . add(25)
a . add(10.8)
a . add(1)
a . add('Hyd')
a . add(25)
a . add(None)
a . add('Hyd')
a . add(1.0)
print(a) #{None, True, 10.8, 25, 'Hyd'}
a . add(10 , 20 , 30) #error
a . add([10,20,30]) #error
```

add()   method

----

1) What does add(x) do ? ---> Inserts 'x' any where in the set becoz set is unordered
2) How many arguments can add() method take ? ---> Single
3) Is set . add(mutable-object) valid ? ---> No becoz set can not hold mutable element
4) In other words, argument of add() method should be immutable object only
5) What does set . add(sequence) do ? --->Inserts sequence any where in the set but not elements of sequence(Like append() method of list class)

# Find outputs (Home work)

```
a = {25 , 10.8 , 'Hyd' , True}
tpl = (10 , 20 , 30)
print(a) # {25, 10.8, 'Hyd', True}
print(id(a)) #adress of a
a . add(tpl)
a . add('Sec')
print(a) # {25, 10.8, 'Hyd', True, (10, 20, 30), 'Sec'}
print(id(a)) #same adress of a
print(len(a)) #6
a . add([100 , 200 , 300]) #erroe:list is mutable
a . add(set()) #error:set is mutable
```

```
a . add({ }) #error:dict is mutable
```

# Find outputs (Home work)

```
s = set()
tpl = (10 , 20 , 15 , 18)
s . add(tpl)
print(s) #{(10, 20, 15, 18)}
print(len(s)) #1
```

# update() method demo program (Home work)

```
tpl = (10 , 20 , 15, 18 , 10 , 20)
s = set()
s . update(tpl)
print(len(s)) #4
print(s) #{10,18,20,15}
s . update(25) #error:int is non sequence
update() method
--------------------
```

1) What does update(sequence) do ? ---> Inserts elements of sequence anywhere in the set but not sequence (Like extend() method of list class)
2) Is update(non-sequnece) valid ? ---> No becoz agument should be sequence only
3) How many arguments can update() method take ? ---> One (or) more

# Find outputs (Home work)

```
a = [10 , 20 , 30]
b = {30 , 40,50 }
c = (50 , 60 , 70)
s = set()
s . update(a , b , c)
print(s) #{10, 20, 30, 40, 50, 60, 70}
print(len(s)) #7
s . add(a , b , c) #error: add() takes exactly one argument (3 given)
```

# Find outputs (Home work)

```
a = set()
a . update('Rama Rao')
print(a) # {'R', 'a', 'm', ' ', 'o'}
print(len(a)) #5
```

```
a . update(3 + 4j , 10.8 , True) #error: non-sequnce type

# copy()  method  demo  program  (Home  work)
a = {10 , 20 , 15 , 18}
print(a) #{10, 18, 20, 15}
b = a . copy()
print(b) #{10, 18, 20, 15}
print(a  is  b) #False
print(a  ==  b) #True
c = a
print(a  is  c) #True
copy()  method
------------------
1) What  does  copy()  method  do ?  ---> Returns  a  new  set  with  same  elements
2) a = b
   What  does  the  statement  do ?  ---> Reference  copy
                                         i.e.  id  is  copied
3) What  is  shallow  clone ?  ---> Reference  copy
    What  is  deep  clone ?  ---> Object  copy

# pop()  method  demo  program  (Home  work)
a = {25 , 10.8 , 'Hyd' , True}
print(a) #{25, 10.8, 'Hyd', True}
print(a . pop()) #25
print(a . pop()) #10.8
print(a . pop()) #Hyd
print(a . pop()) #True
#print(a . pop()) #Error  (Set  is  empty)
print(a) #set()
b = {10 , 20 , 30 , 40}
print(b . pop(2)) #error:  pop()  takes  no  arguments  (1  given)
'''
pop()  method
----------------
1) What  does  pop(No-args)  method  do ?  ---
> Removes  first  element  of  the  set  and  returns  the  deleted  element
2) What  does  emptyset . pop()  do ?  ---> Throws  error
3) Is  set . pop(index)  valid ?  ---> No  becoz  set  is  not  indexed
```

4) How many arguments can pop() method take ? ---> Zero

# remove() method demo program (Home work)

```
a = {25 , 10.8 , 'Hyd' , True}
print(a) #{25, 10.8, 'Hyd', True}
a . remove('Hyd')
print(a) #{25, 10.8, True}
a . remove('Sec') #error
```

remove() method

----

1) What does remove(x) do ? ---> Removes 'x' from the set
2) What does remove(Invalid-element) do ? ---> Throws error
3) What is the argument of remove() method ? ---> Element to be removed

# discard() method demo program (Home work)

```
a = {25 , 10.8 , 'Hyd' , True}
print(a) #{25, 10.8, 'Hyd', True}
a . discard('Hyd')
print(a) #{25, 10.8, True}
a . discard('Sec')
print(a) #{25, 10.8, True}
a . remove('Sec') #error
```

discard() method

----

1) What does discard(x) do ? ---> Removes 'x' from the set
2) What does discard(Invalid-element) do ? ---> Nothing
3) In other words, discard(invalid-element) does not raise error nor deletion

# clear() method demo program (Home work)

```
a = {10 , 20 , 15 , 18}
print(a) #{10, 20, 15, 18}
a . clear()
print(a) #set()
print(len(a)) #0
```

clear() method

What does clear() method do ? ---
> Removes all the elements of set and set becomes empty

# Find outputs (Home work)

```
a = {10 , 20 , 30 , 40}
b = [30 , 40 , 50 , 60]
print(a . union(b)) #{10, 20, 30, 40, 50, 60}
print(a | b) #error:list and set can't use |
print(b . union(a))  #error:list has no union method
print(a + b) #error: set and list can't use +
```