H.T. No. ..... Date .....

```
# Find outputs (Home work)
a = [25, 10.8, 'Hyd', True, 3+4j, None, 'Hyd', 25]
Print(a) # 25, 10.8, 'Hyd', True, (3+4j), None,
           'Hyd', 25]
Print(*a) # 25 10.8 Hyd True (3+4j) None
            Hyd 25.
Print (type(a)) # <class 'list'>
Print (id(a)) # some memory address, changes
                every run.
Print (len(a)) = 8
a [2] = 'sec' # printing the modified list
Print(a) # [25, 10.8, 'sec', True, (3+4j),
           None, 'Hyd', 25]
print (a[2:5]) # ['sec', True, (3+4j)]
```

```
# append() and remove() methods (Home work)
a = [] # printing the empty list
Print(a) # []
a. append (25) # Appending float 10.8
a. append (10.8) # Appending string 'Hyd'
a. append ('Hyd') # Appending boolean True
a. append (True) # printing the list after
                   appending elements
```

H.T. No. .....

Date .....

```
print(a) # [25, 10.8, 'Hyd', True]
a.remove('Hyd') # printing the list after removing 'Hyd'
print(a) # [25, 10.8, True]
a.remove('25') # Error.
print(a)
```

Find outputs (Home work).

```
a = [25, 10.8, 'Hyd']
print(a) # [25, 10.8, 'Hyd']
print(id(a)) # some memory address
print(a*3) # 25, 10.8, 'Hyd', 25, 10.8, 'Hyd',
             25, 10.8, 'Hyd'
print(a*2) # 25, 10.8, 'Hyd', 25, 10.8, 'Hyd'
print(a*1) # [25, 10.8, 'Hyd']
print(a*0) # Empty list.
print(a*-1) # Empty list
Print(a*4.0) # Error
a = a*3 # printing the updated list
print(a) # 25, 10.8, 'Hyd', 25, 10.8, 'Hyd',
           25, 10.8, 'Hyd']
print(id(a)) # New memory address
a = [25]
Print(a*a) # Error.
```

Page No.

H.T. No. ..... Date .....

```
print ('25'[0])    # output: 2
print (True[1])    # Error 'bool' object is not
                     subscriptable
print ('True'[1])  # output: r
```

```
# Find outputs (Home work)
a = 'Hyd' # string assigned to variable 'a'
print (a*3)  # HydHydHyd
print (a*2)  # HydHyd
print (a*1)  # Hyd
print (a*0)  # ''
print (a*-1) # ''
print (25*3)  # 75
print ('25'*3) # 252525
print ('25'*4.0) # Error 'TypeError'
print (3*'Hyd') # HydHydHyd
print ('25'*True) # 25
```

```
# Find outputs (Home work).
a = 'Hyd'  # Assigning a string to variable 'a'.
print (a, id(a)) # Hyd.
a = a*3  # Repeating the string 3 times and
           reassigning it to 'a'.
print (a, id(a)) # a = 'HydHydHyd'
                 # HydHydHyd.
```

```
print(list[:]) # same as above, gives full list
print(list[::-1]) # reverse list ['Hyd', 10.8, None,
True, 'Hyd', (3+4j), 10.8, 25]
print(list[::2]) # [25, 3+4j, True, 10.8]
print(list[1::2]) # every 2nd element from index1
                  [10.8, 'Hyd', None, 'Hyd']
print(list[::-2]) # reverse and step -2 ['Hyd',
                  None, 'Hyd', 10.8]
print(list[-2::-2]) # [10.8, 'Hyd', 10.8, 25]
print(list[1:4]) # [10.8, 3+4j, 'Hyd']
print(list[-4:-1]) # [True, None, 10.8]
print(list[3:-3]) # ['Hyd']
print(list[2:-5]) # Empty.
print(list[-1:-5]) # slicing backwards but
    no step specified → returns []
```

```
# Find outputs (Home work)
#  0    1    2    3    4    5    6    7
list = [25, 10.8, 3+4j, 'Hyd', True, None, 10.8,
                                    'Hyd']
x, y = list[3:5] # ['Hyd', True] unpacking: x=
                                    'Hyd', y=True

print('x:', x) # Hyd
print('y:', y) # True.
for x in list[2:7]: # [3+4j, 'Hyd', True, None,
                      10.8].

    print(x)
```

H.T No. Date

```
# Find outputs (Home work)
#          0   1   2   3   4
a = [10, 20, 30, 40, 50]
print(a, id(a)) # [10,20,30,40,50] < same-id
a[1:4] = [60,70] # a[1:4] refers to [20,30,4
Replace with [60,70], the list change length
   from 5 to 4. [10,60,70,50]
print(a, id(a)) # [10,60,70,50]
                  < same-id-as-before>
a[2:4] = [100, 200, 300] # New a
                           [10,60,100,200,300]
print(a, id(a)) # [10,60,100, 200, 300]
                  < same-id-as-before>
```

```
# Find outputs (Home work).
a = [25] # This will cause an index error
print(a[1:]) # ~~Empty~~ This will cause an In
print[a[1:]) # Empty.
```

```
# Find outputs (Home work).
a = (25) # Not a tuple
b = 25 # Integer
c = 25 # Integer
d = (25,) # Tuple (single-element tuple
requires a trailing comma)
```

```
print(type(a))  # <class 'int'>
print(type(b))  # <class 'int'>
print(type(c))  # <class 'int'>
print(type(d))  # <class 'tuple'>
print(a*4)      # 100
print(b*4)      # 100   -> 25*4=100
print(c*4)      # 100   -> 25*4=100
print(d*4)      # (25,25,25,25) -> Tuple repetition
```

```
# tuple() function demo program
a=tuple('Hyd')  # 'Hyd' is a string
print(a) # ('H', 'y', 'd')
print(type(a)) # <class 'tuple'>
print(len(a)) # 3.
b=[10,20,15,18]
print(tuple(b))#  (10,20,15,18)
print(tuple(range(5))) # (0,1,2,3,4)
print(tuple(25)) # Error.
```

```
# Find Outputs (Home work)
a=()   # empty tuple
print(type(a)) # <class 'tuple'>
print(a)  # () (empty tuple)
print(len(a)) # 0.
b=tuple()  # b is also an empty tuple
print(b)  # () (empty tuple)
print(len(b)) # 0
```

H.T. No. Date

```
# tricky program.
# find outputs.
a = (10, 20, 30) # a is a tuple.
print(a) # (10, 20, 30)
Print(id(a)) # this points the memory address
a = a*2 # Repeat the tuple twice
                (10, 20, 30, 10, 20, 30)
print(a) # (10, 20, 30, 10, 20, 30)
Print(id(a)) # A new ID.
```

```
# set() function demo program.
a = set('Rama&AO')
Print(a) # {' ', 'R', 'A', 'a', 'm', 'O', '&'}
print(len(a)) # length=7.
Print(set([10, 20, 15, 20])) # {10, 20, 15}
print(set((25, 10.8, 'Hyd', 10.8))) # {25, 10.8,
Print(set(range(10, 20, 3))) # {10, 13, 16, 19}
print(set(25)) # Error
Print(set([25, 10.8, [], 'Hyd'])) # Error
```

```
# Find Outputs (Home work)
a = []
b = ()
c = {}
d = set()
print(type(a)) # <class 'list'>
```

```
print(type(b)) # <class 'tuple'>
print(type(c)) # <class 'dict'>
print(type(d)) # <class 'set'>
print(a) → []
print(b) → ()
print(c) → {}
print(d) → ()
```

Tricky program
add() and remove() methods (Home work)

```
a=set()
a.add(25)
a.add(10.8)
a.add('Hyd')
a.add(True)
a.add(None)
a.add('Hyd') # Duplicate, ignored.
a.add(1)     # 1 is already presents as True
print(a) # {None, 10.8, 'Hyd', 1, 25}.
print(len(a)) # 5
a.remove(25).
print(a) # {None, 10.8, 'Hyd', 1}
a.append(100) # Error
a.add(set()) # Allowed
a.add(1) # Allowed
a.add(0) # Allowed
```

H.T. No. .....

Date :

```
a.add([])  # Error.
print(a)
a.add({})  # Error
```

```
# How to print set in two different ways
a = {25, True, 'Hyd', 10.8}.
print('set with print function :')
print(a)
print('How to print set') # using for loop
print('Iterate through set with for loop:')
for item in a:
    print(item)
```