

NAME : AKULA SHARATH CHANDRA

BATCH: DATA ENGINEERING

DATE : 31-01-24

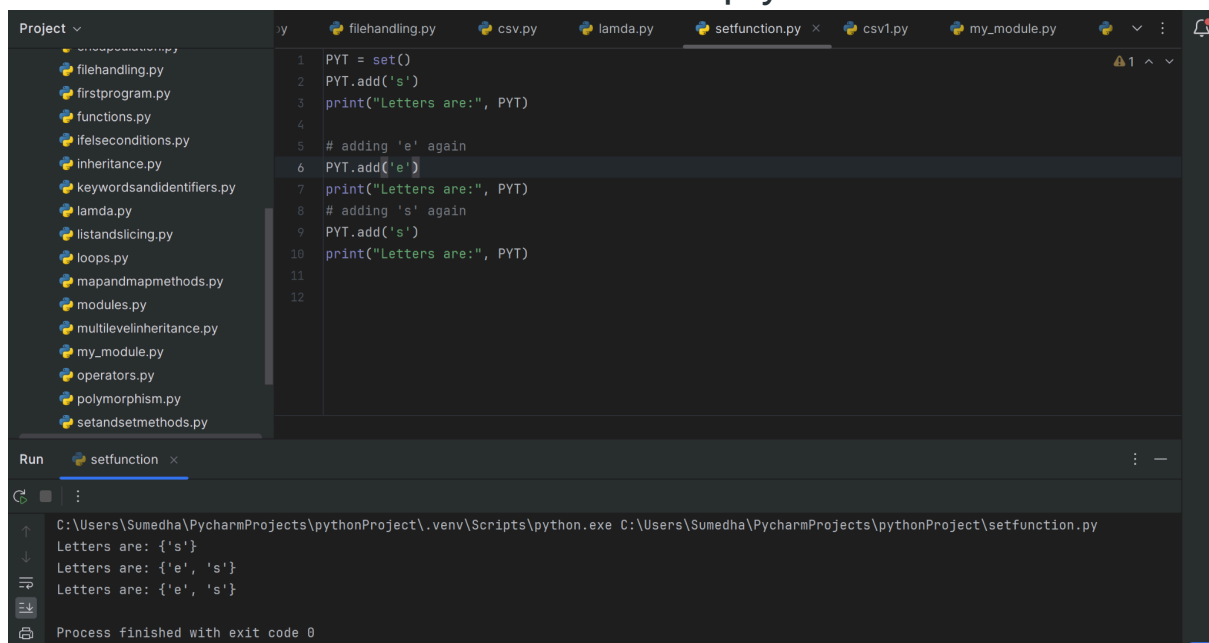
TOPICS : SET() METHODS,

1) Python Set add() Method: The Python set add() method adds a given element to a set if the element is not present in the set in python.

Syntax of Set add()

Syntax: set.add(element)

EXAMPLE: Add Element to an Empty set



The screenshot displays the PyCharm IDE interface. The left sidebar shows a project tree with various Python files. The main editor window is open to a file named 'setfunction.py', which contains the following code:

```
1 PYT = set()
2 PYT.add('s')
3 print("Letters are:", PYT)
4
5 # adding 'e' again
6 PYT.add('e')
7 print("Letters are:", PYT)
8 # adding 's' again
9 PYT.add('s')
10 print("Letters are:", PYT)
11
12
```

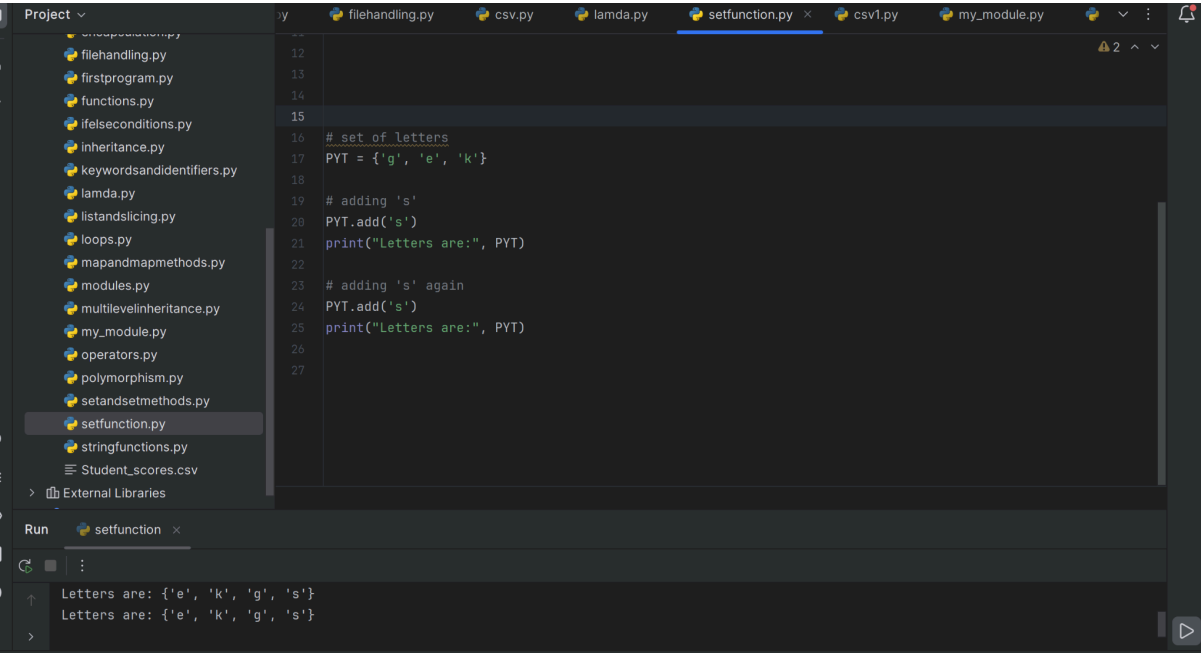
Below the editor, the 'Run' tab is active, showing the execution output for 'setfunction'. The output is as follows:

```
C:\Users\Sumedha\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\Sumedha\PycharmProjects\pythonProject\setfunction.py
Letters are: {'s'}
Letters are: {'e', 's'}
Letters are: {'e', 's'}
```

At the bottom of the Run tab, it states 'Process finished with exit code 0'.

ii) Add a new element to a Python set:

It is used to add a new element to the set if it is not existing in a set.



```
12
13
14
15
16 # set of letters
17 PYT = {'g', 'e', 'k'}
18
19 # adding 's'
20 PYT.add('s')
21 print("Letters are:", PYT)
22
23 # adding 's' again
24 PYT.add('s')
25 print("Letters are:", PYT)
26
27
```

Run setfunction x

Letters are: {'e', 'k', 'g', 's'}

Letters are: {'e', 'k', 'g', 's'}

iii) Add element in a set that already exists: It is used to add an existing element to the set if it is existing in the python set and check if it gets added or not.



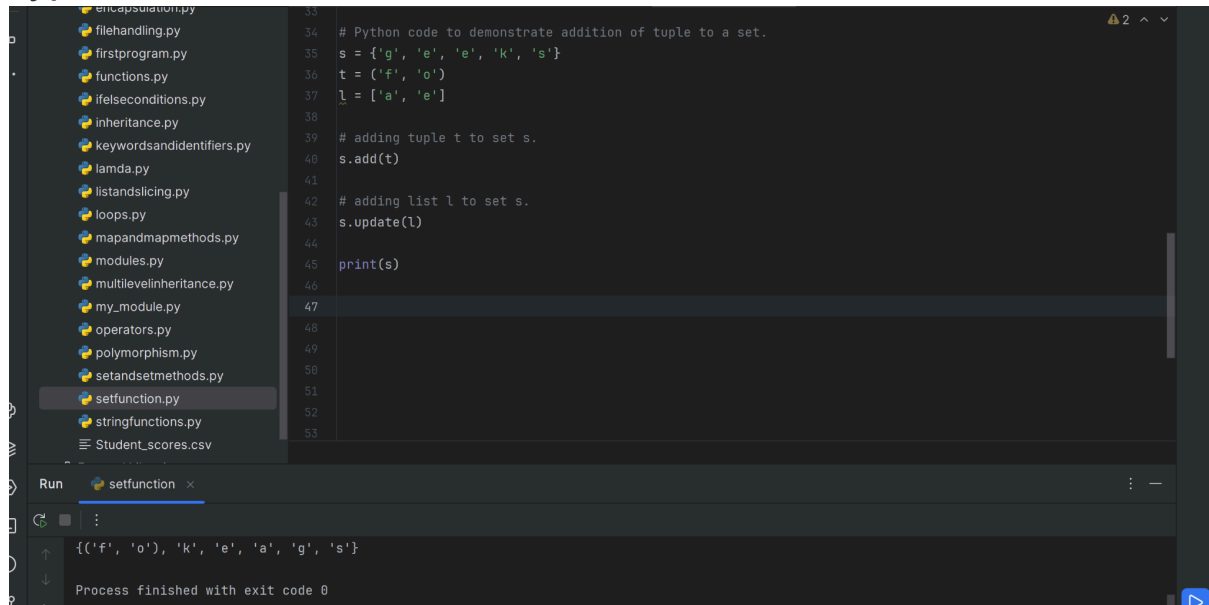
```
22
23 # set of letters
24 PYT = {6, 0, 4}
25
26 # adding 1
27 PYT.add(1)
28 print('Letters are:', PYT)
29
30 # adding 0
31 PYT.add(0)
32 print('Letters are:', PYT)
33
34
35
36
37
38
39
40
41
42
```

Run setfunction x

Letters are: {0, 1, 4, 6}

Letters are: {0, 1, 4, 6}

iv) **Adding any iterable to a set:** We can add any Python iterable to a set using Python add or python update function, if we try to add a list using the add function we get an unhashable Type error.



```
33
34 # Python code to demonstrate addition of tuple to a set.
35 s = {'g', 'e', 'e', 'k', 's'}
36 t = ('f', 'o')
37 l = ['a', 'e']
38
39 # adding tuple t to set s.
40 s.add(t)
41
42 # adding list l to set s.
43 s.update(l)
44
45 print(s)
46
47
48
49
50
51
52
53
```

Run setfunction x

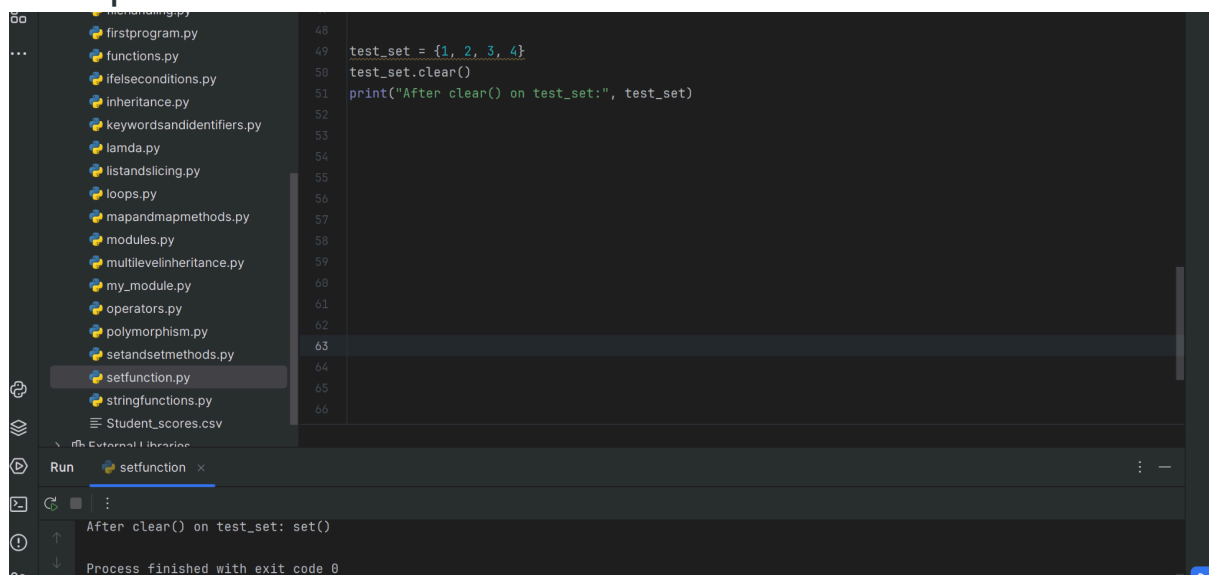
{('f', 'o'), 'k', 'e', 'a', 'g', 's'}

Process finished with exit code 0

v) Python Set clear() Method Syntax:

Syntax: set.clear()

example:



```
48
49 test_set = {1, 2, 3, 4}
50 test_set.clear()
51 print("After clear() on test_set:", test_set)
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
```

Run setfunction x

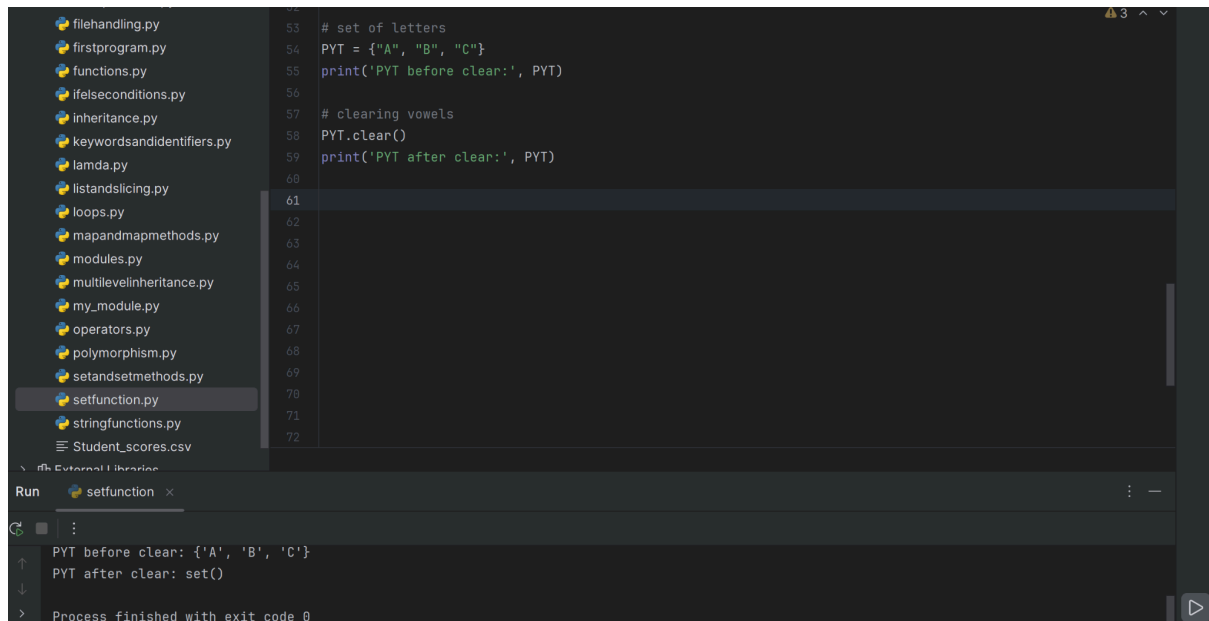
After clear() on test_set: set()

Process finished with exit code 0

The time complexity of the clear() method on a set in Python is $O(n)$, where n is the number of elements in the set.

The auxiliary space complexity of the clear() method is also $O(1)$, as it only needs to delete the existing elements in the set and not allocate any new memory.

vi) Python Set clear() Method on a Set of Strings:



```
52
53 # set of letters
54 PYT = {"A", "B", "C"}
55 print('PYT before clear:', PYT)
56
57 # clearing vowels
58 PYT.clear()
59 print('PYT after clear:', PYT)
60
61
62
63
64
65
66
67
68
69
70
71
72
```

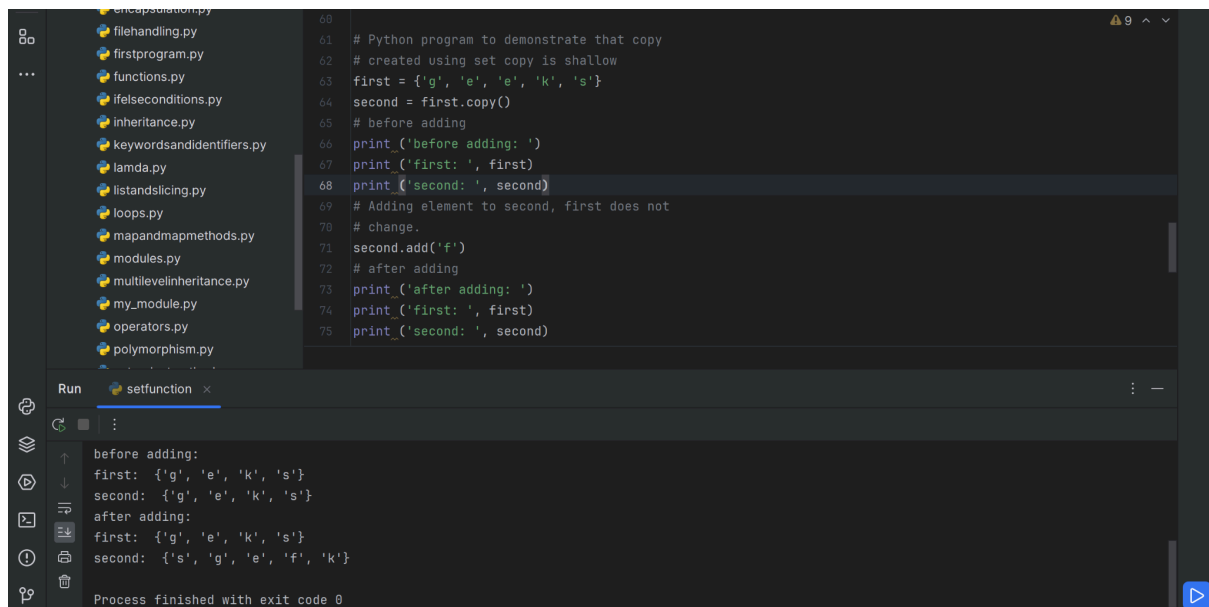
Run setfunction x

PYT before clear: {'A', 'B', 'C'}

PYT after clear: set()

Process finished with exit code 0

vii)Shallow Copy Example :A shallow copy creates a new object, but does not create new objects for the elements contained in the original object. Instead, it copies references to the objects. In Python, you can use the copy module to create shallow copies.



```
60
61 # Python program to demonstrate that copy
62 # created using set copy is shallow
63 first = {'g', 'e', 'k', 's'}
64 second = first.copy()
65 # before adding
66 print('before adding: ')
67 print('first: ', first)
68 print('second: ', second)
69 # Adding element to second, first does not
70 # change.
71 second.add('f')
72 # after adding
73 print('after adding: ')
74 print('first: ', first)
75 print('second: ', second)
```

Run setfunction x

before adding:

first: {'g', 'e', 'k', 's'}

second: {'g', 'e', 'k', 's'}

after adding:

first: {'g', 'e', 'k', 's'}

second: {'s', 'g', 'e', 'f', 'k'}

Process finished with exit code 0

Time complexity : $O(1)$

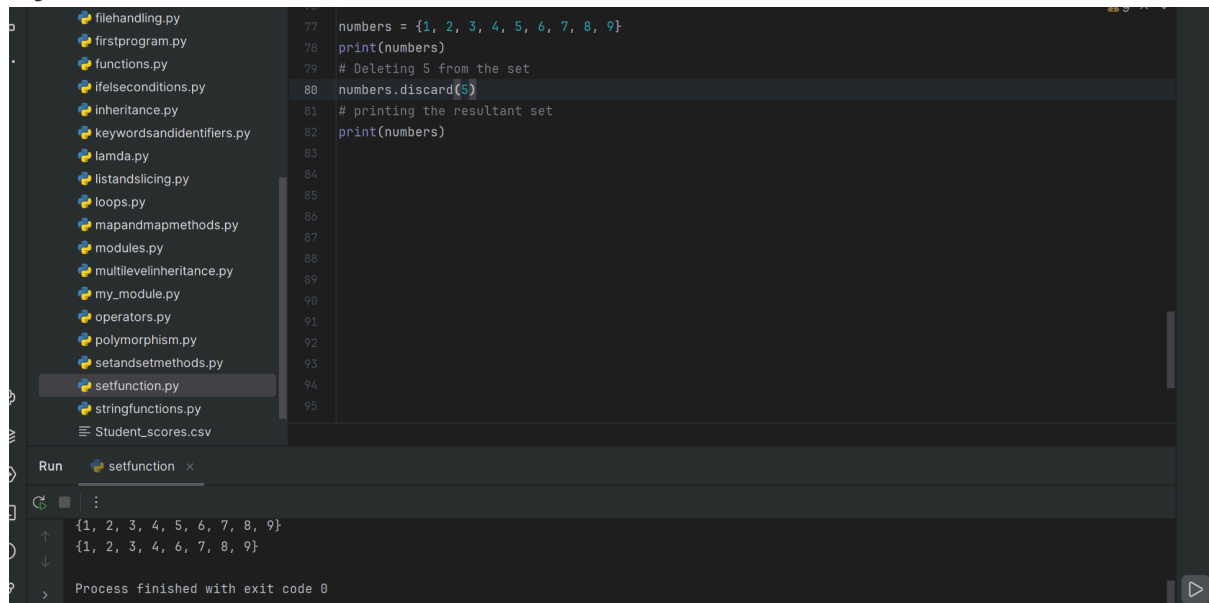
space complexity: $O(n)$

viii)**Python Set discard():**Python discard() is a built-in method to remove elements from the set.

The discard() method takes exactly one argument. This method does not return any value.

Python Set discard() Syntax: set.discard(element)

Example:Discard() an item from a set that is present in the Python Set

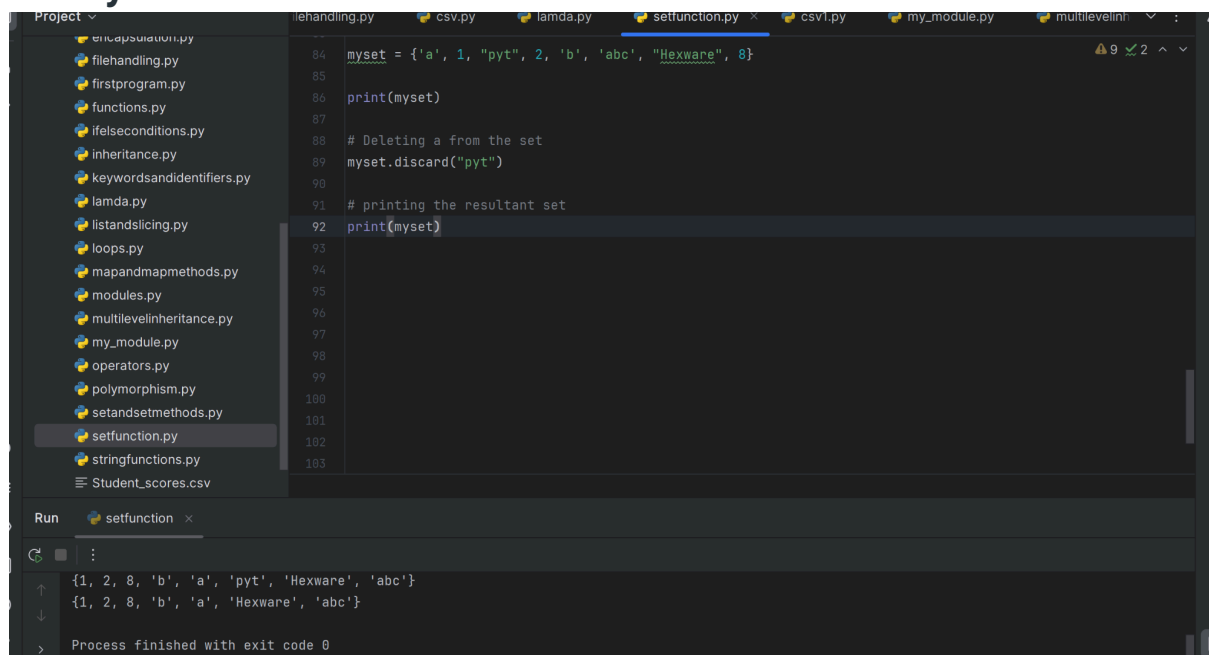


```
77 numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9}
78 print(numbers)
79 # Deleting 5 from the set
80 numbers.discard(5)
81 # printing the resultant set
82 print(numbers)
83
84
85
86
87
88
89
90
91
92
93
94
95
```

Run setfunction x

```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
{1, 2, 3, 4, 6, 7, 8, 9}
Process finished with exit code 0
```

Example 2:Discard() a String item from a set that is present in the Python Set

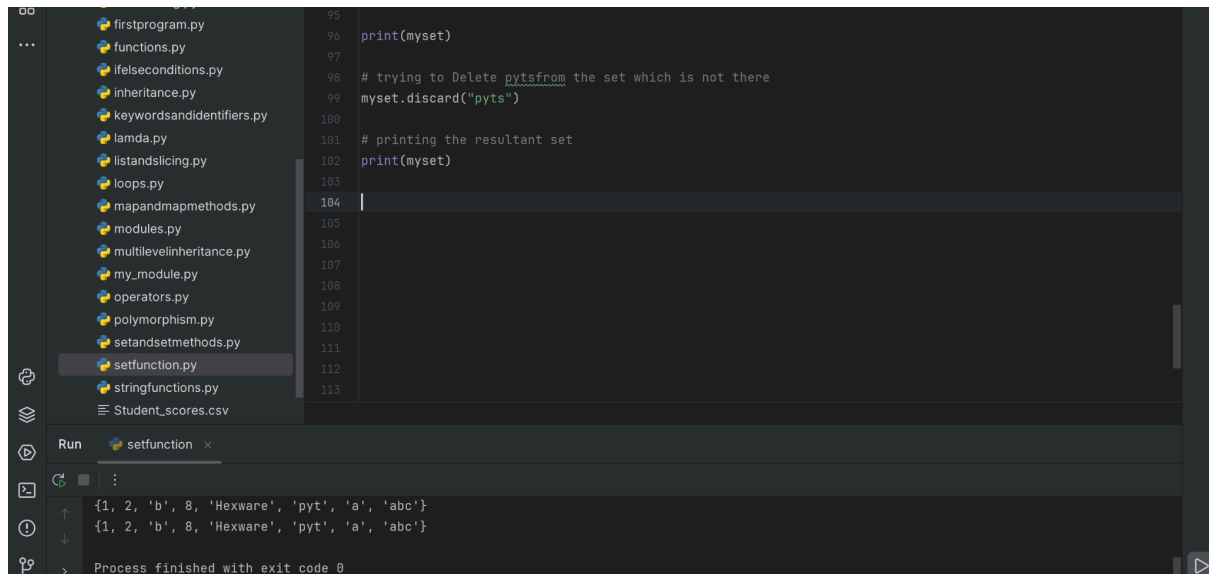


```
84 myset = {'a', 1, "pyt", 2, 'b', 'abc', "Hexware", 8}
85
86 print(myset)
87
88 # Deleting a from the set
89 myset.discard("pyt")
90
91 # printing the resultant set
92 print(myset)
93
94
95
96
97
98
99
100
101
102
103
```

Run setfunction x

```
{1, 2, 8, 'b', 'a', 'pyt', 'Hexware', 'abc'}
{1, 2, 8, 'b', 'a', 'Hexware', 'abc'}
Process finished with exit code 0
```

Example 3: Discard() String item from a set that is not present in the Python Set



```
95 print(myset)
96
97
98 # trying to Delete pyts from the set which is not there
99 myset.discard("pyts")
100
101 # printing the resultant set
102 print(myset)
103
104
105
106
107
108
109
110
111
112
113
```

Run setfunction x

```
{1, 2, 'b', 8, 'Hexware', 'pyt', 'a', 'abc'}
{1, 2, 'b', 8, 'Hexware', 'pyt', 'a', 'abc'}
```

Process finished with exit code 0