

Raul Rodriguez

```
Lab7_1.py:7: ⚠ Intersect function
1  '''Exercise 1
2  Create a list of numbers from 10 to 20 and out of the list create two sets where the first set, set
3  A, contains the numbers 10 to 20 (inclusive) and the second set, set B, contains the odd numbers
4  of the list (use set comprehension). Ask the user whether they wish to add, remove, perform
5  union, intersection, difference, symmetric difference, or disjoint between the two sets. For
6  adding or removing elements, ask user on which set they wish to carry out the operation. Use
7  functions for all seven operations'''
8  def addFunction(sett):
9      num=int(input("Enter the number you want to add to the set: "))
10     sett.add(num)
11     print(sett)
12 def removeFunction(sett):
13     num=int(input("Enter the number you want to add to the set: "))
14     sett.remove(num)
15     print(sett)
16 def unionFunction(set1,oddSet):
17     print(set1 | oddSet)
18 def intersectFunction(set1,oddSet):
19     print([set1 & oddSet])
20 def differenceFunction(set1,oddSet):
21     print(set1 - oddSet)
22 def symmetricDiffFunction(set1,oddSet):
23     print(set1 ^ oddSet)
24 def disjointFunction(set1,oddSet):
25     print(set1.isdisjoint(oddSet))
26 li=[10,11,12,13,14,15,16,17,18,19,20]
27 set1=set(li)
28 oddSet={item for item in li if item %2 !=0}
29 print(set1)
30 print(oddSet)
31 print("Choose your operation on the provided sets: ")
32 choice=int(input("1=add, 2=remove, 3=union, 4=intersection, 5=difference, 6=symmetric difference, or 7=disjoint: "))
33 if choice==1:
34     choice2=int(input("Would you like to perform operation on original set, or odd set? 1=original, 2=odd: "))
35     if choice2==1:
36         addFunction(set1)
37     elif choice2==2:
38         addFunction(oddSet)
39 elif choice==2:
40     choice2=int(input("Would you like to perform operation on original set, or odd set? 1=original, 2=odd: "))
41     if choice2==1:
42         removeFunction(set1)
43     elif choice2==2:
44         removeFunction(oddSet)
45 elif choice==3:
46     unionFunction(set1,oddSet)
47 elif choice==4:
48     intersectFunction(set1,oddSet)
49 elif choice==5:
50     differenceFunction(set1,oddSet)
51 elif choice==6:
52     symmetricDiffFunction(set1,oddSet)
53 elif choice==7:
54     disjointFunction(set1,oddSet)
```

```

● s/raulrodriguez/Documents/WorkSpaceVSPython/Lab7_1.py
{10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
{11, 13, 15, 17, 19}
Choose your operation on the provided sets:
1=add, 2=remove, 3=union, 4=intersection, 5=difference, 6=symmetric difference, or 7=disjoint: 4
{11, 13, 15, 17, 19}
○ raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % █

```

```

● Lab7_2.py > ...
1  '''Exercise 2
2  Create a 2D list with 3 rows and 2 columns and various values in it. Ask user which element
3  of the array they wish to modify given the row and column numbers as well as the new value of
4  the element; print list'''
5  a=[[5,2],[3,4],[7,9]]
6  print(a)
7  row=int(input("Enter row of value you want to modify: "))
8  col=int(input("Enter column of number you want to modify: "))
9  num=int(input("Enter new value: "))
10 a[row][col]=num
11 print(a)

```

```

● raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /Users/raulrodriguez/Documents/WorkSpaceVSPython/Lab7_2.py
[[5, 2], [3, 4], [7, 9]]
Enter row of value you want to modify: 1
Enter column of number you want to modify: 1
Enter new value: 5
[[5, 2], [3, 5], [7, 9]]
○ raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % █

```

```

● Lab7_3.py > ...
1  '''Exercise 3
2  Ask user to enter the min and max values of a function (use int), e.g., -10 to 10. Create a
3  range of numbers from min to max and assign it to a variable named x. Compute the absolute
4  value of the range using a list comprehension using the format: [expression1 if condition else
5  expression2 for item in iterable] and assign it to variable named y. Plot x, y and add a title to
6  your plot as well as labels for x and y axes'''
7  from matplotlib import pyplot as plt
8  min=int(input("Enter minimum of range: "))
9  max=int(input("Enter maximum of range: "))
10 x=range(min,max+1)
11 print(list(x))
12 y=[c if c>0 else -c for c in x]
13 print(y)
14 plt.plot(x,y)
15 plt.xlabel('x-axis')
16 plt.ylabel('y-axis')
17 plt.title("Graph of |x|")
18 plt.show()

```

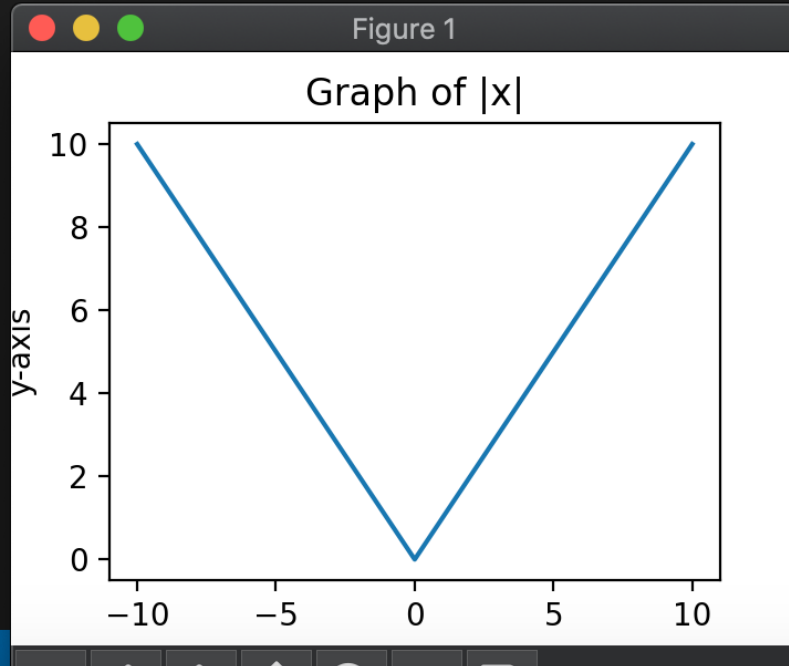
```

raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/bin/python3 /U
Enter minimum of range: -10
Enter maximum of range: 10
[-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
/Users/raulrodriguez/Documents/WorkSpaceVSPython/Lab7_3.py:14: MatplotlibDeprec
ved two minor releases later. Use callbacks.process('resize_event', ResizeEvent
plt.plot(x,y)

```

```

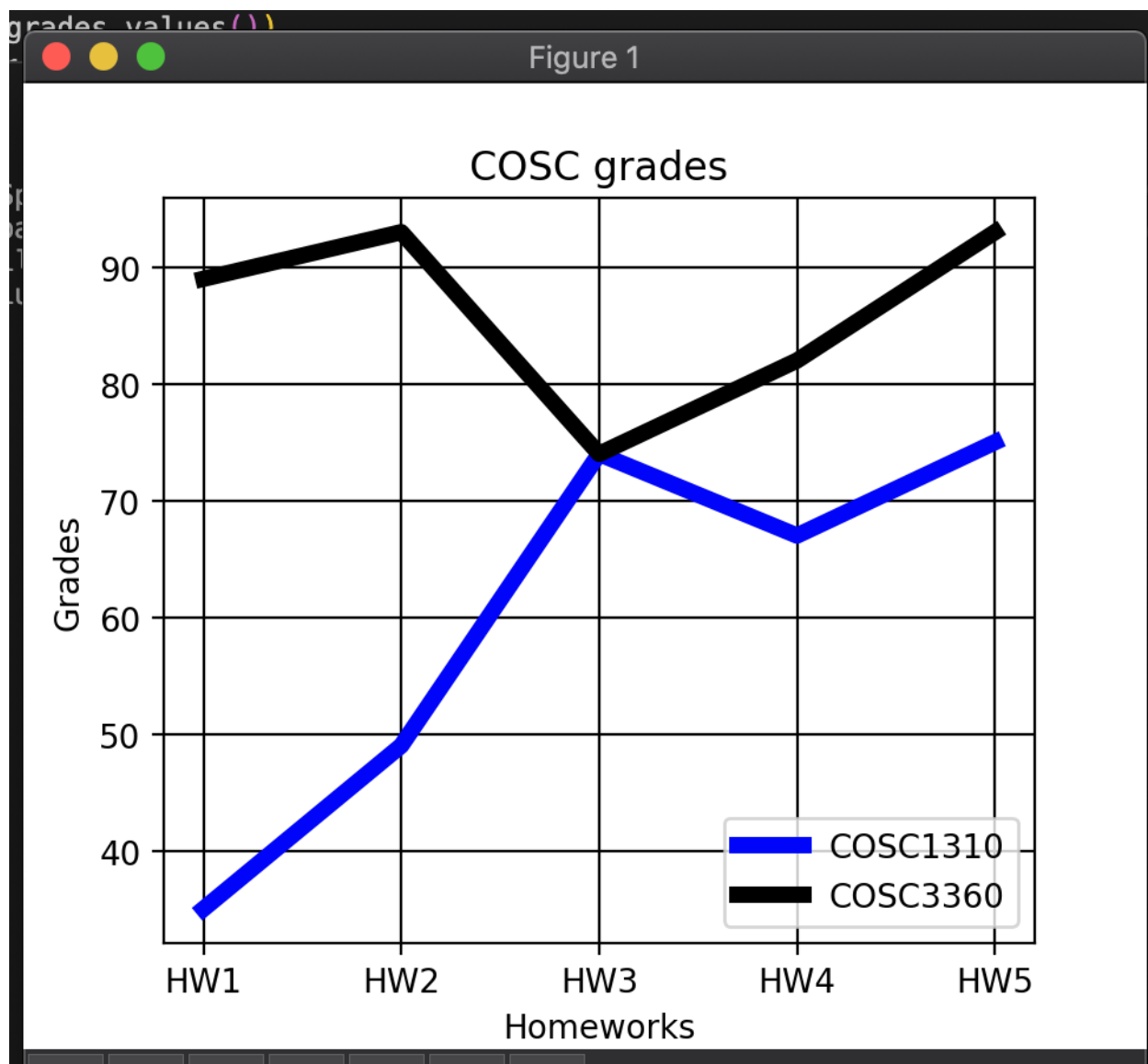

```



```

1  '''Exercise 4
2  Using the following two dictionaries:
3  COSC1310grades = {"HW1" : 35, "HW2" : 49, "HW3" : 74, "HW4" : 67, "HW5" : 75}
4  COSC3360grades = {"HW1" : 89, "HW2" : 93, "HW3" : 74, "HW4" : 82, "HW5" : 93}
5  create a single graph, with two plots on it, one for each dictionary. The x and y values
6  correspond to the key and value of each dictionaries. You can use the built-in functions to extract
7  the keys and values from the dictionaries. Add labels, color, linewidth, and legend to your
8  graph'''
9  from matplotlib import pyplot as plt
10 COSC1310grades = {"HW1" : 35, "HW2" : 49, "HW3" : 74, "HW4" : 67, "HW5" : 75}
11 COSC3360grades = {"HW1" : 89, "HW2" : 93, "HW3" : 74, "HW4" : 82, "HW5" : 93}
12 COSC1310_keys=list(COSC1310grades.keys())
13 COSC1310_values=list(COSC1310grades.values())
14 COSC3360_keys=list(COSC3360grades.keys())
15 COSC3360_values=list(COSC3360grades.values())
16 plt.plot(COSC1310_keys,COSC1310_values,label='COSC1310',linewidth=5,color='b')
17 plt.plot(COSC3360_keys,COSC3360_values,label='COSC3360',linewidth=5,color='k')
18 plt.xlabel('Homeworks')
19 plt.ylabel('Grades')
20 plt.title("COSC grades")
21 plt.legend()
22 plt.grid(True,color='k')
23 plt.show()

```



Lab7_5.py > ...

```
1  '''Exercise 5
2  Same as in Ex. 4, create two graphs with one plot each. Use the subplots function'''
3  from matplotlib import pyplot as plt
4  COSC1310grades = {"HW1" : 35, "HW2" : 49, "HW3" : 74, "HW4" : 67, "HW5" : 75}
5  COSC3360grades = {"HW1" : 89, "HW2" : 93, "HW3" : 74, "HW4" : 82, "HW5" : 93}
6  COSC1310_keys=list(COSC1310grades.keys())
7  COSC1310_values=list(COSC1310grades.values())
8  COSC3360_keys=list(COSC3360grades.keys())
9  COSC3360_values=list(COSC3360grades.values())
10 fig, ax=plt.subplots(nrows=1,ncols=2)
11 ax[0].plot(COSC1310_keys,COSC1310_values,label='COSC1310',linewidth=3,color='b')
12 ax[1].plot(COSC3360_keys,COSC3360_values,label='COSC3360',linewidth=3,color='k')
13 ax[0].set_title('COSC1310')
14 ax[1].set_title('COSC3360')
15 ax[0].set_xlabel('Homeworks')
16 ax[0].set_ylabel('Grades')
17 ax[1].set_xlabel('Homeworks')
18 ax[1].set_ylabel('Grades')
19 ax[0].grid(True,color='k')
20 ax[1].grid(True,color='k')
21 ax[0].set_ylim(0,100)
22 ax[1].set_ylim(0,100)
23 ax[0].legend()
24 ax[1].legend()
25 plt.show()
```

