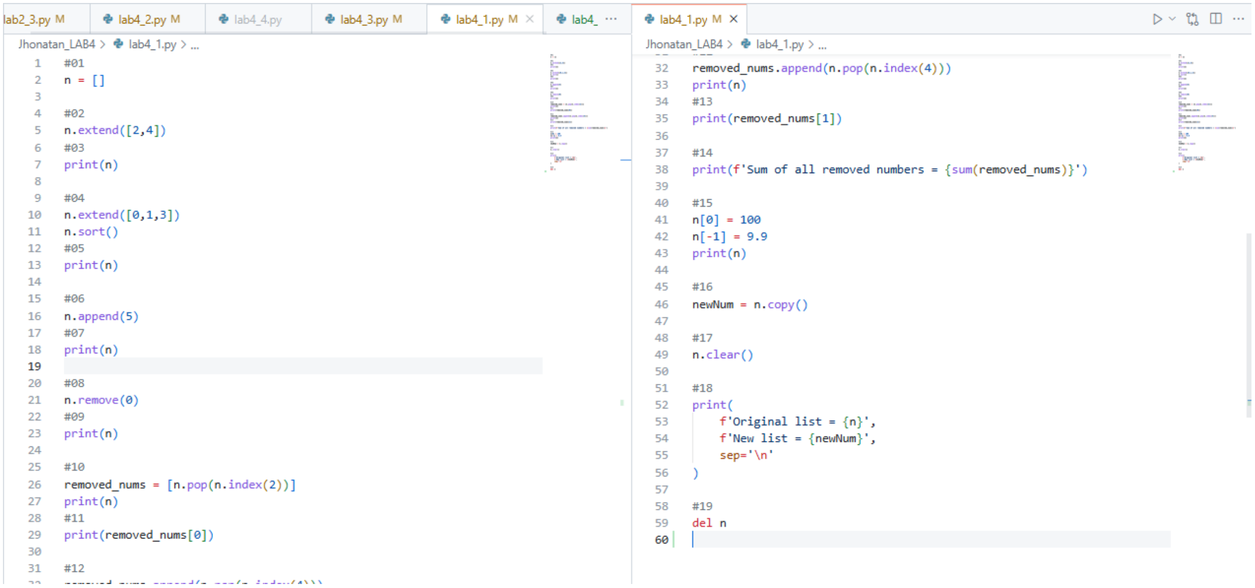
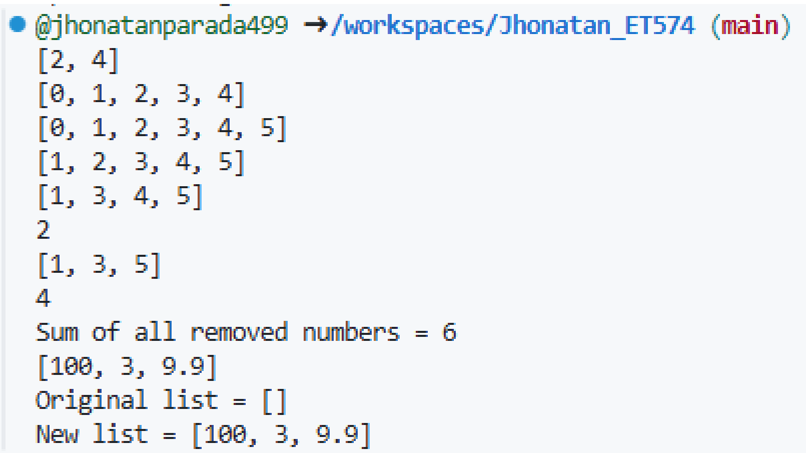
Lab 4 Summary

Jhonatan Parada Torres

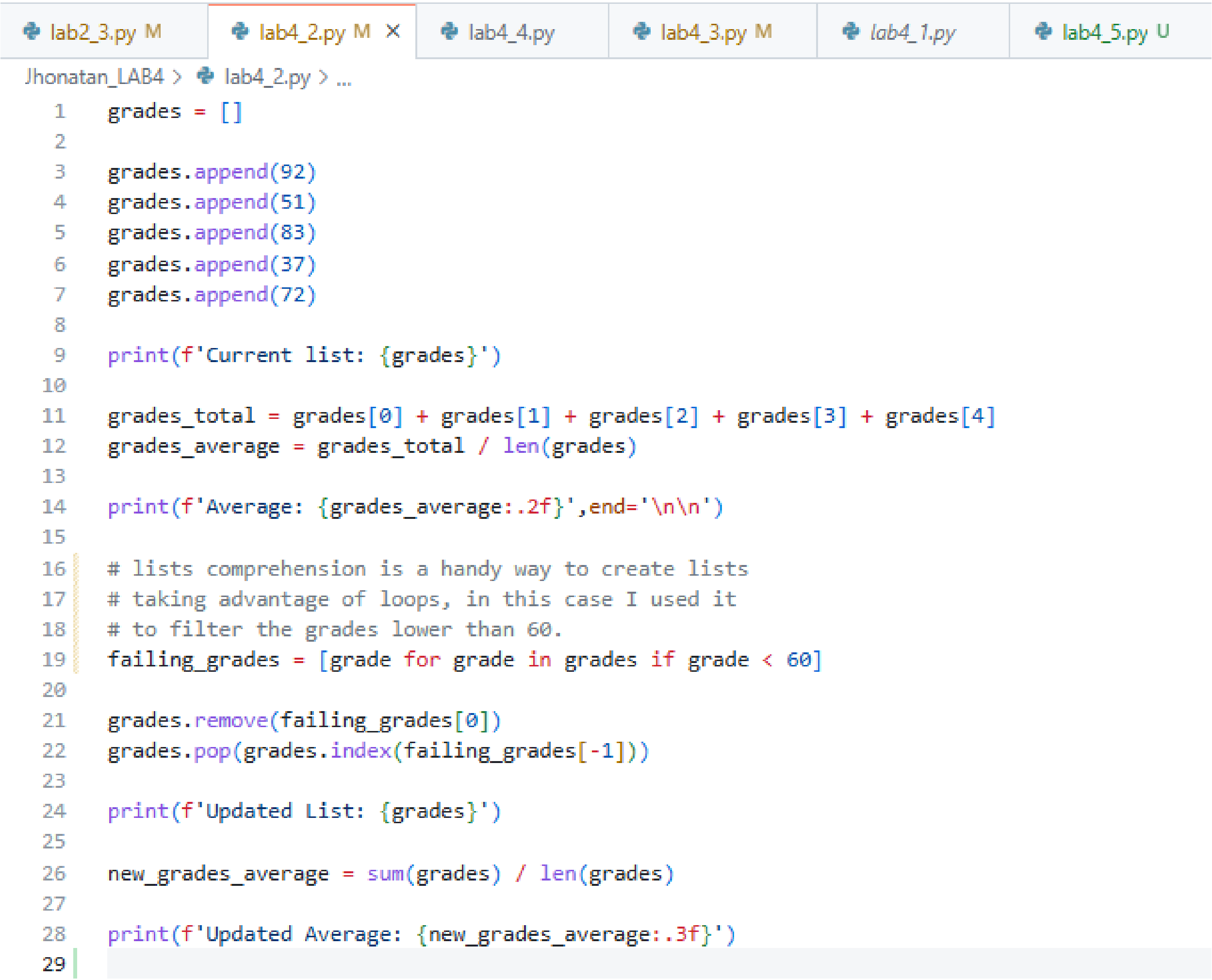
1.



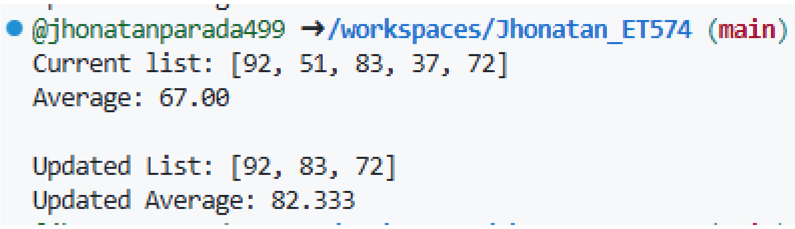
```python
#01
n = []

#02
n.extend([2,4])
#03
print(n)

#04
n.extend([0,1,3])
n.sort()
#05
print(n)

#06
n.append(5)
#07
print(n)

#08
n.remove(0)
#09
print(n)

#10
removed_nums = [n.pop(n.index(2))]
print(n)
#11
print(removed_nums[0])

#12
removed_nums.append(n.pop(n.index(4)))
print(n)
#13
print(removed_nums[1])

#14
print(f'Sum of all removed numbers = {sum(removed_nums)}')

#15
n[0] = 100
n[-1] = 9.9
print(n)

#16
newNum = n.copy()

#17
n.clear()

#18
print(
    f'Original list = {n}',
    f'New list = {newNum}',
    sep='\n'
)

#19
del n

```



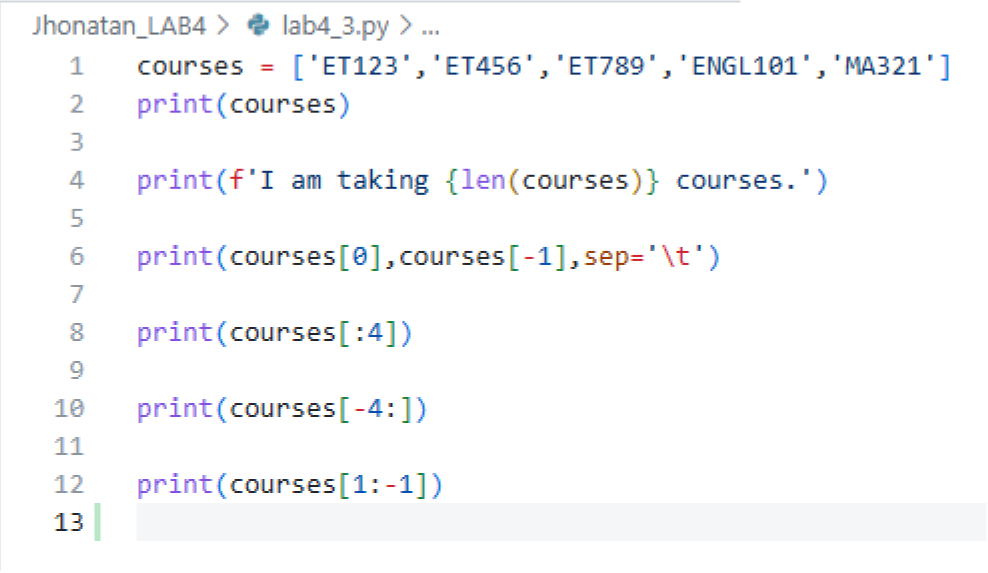
```
● @jhonatanparada499 →/workspaces/Jhonatan_ET574 (main)
  [2, 4]
  [0, 1, 2, 3, 4]
  [0, 1, 2, 3, 4, 5]
  [1, 2, 3, 4, 5]
  [1, 3, 4, 5]
  2
  [1, 3, 5]
  4
  Sum of all removed numbers = 6
  [100, 3, 9.9]
  Original list = []
  New list = [100, 3, 9.9]
```



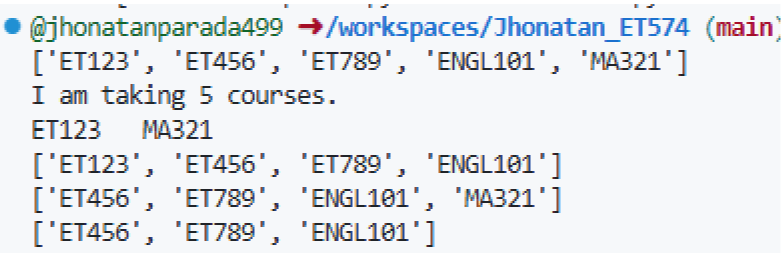
```python
grades = []

grades.append(92)
grades.append(51)
grades.append(83)
grades.append(37)
grades.append(72)

print(f'Current list: {grades}')

grades_total = grades[0] + grades[1] + grades[2] + grades[3] + grades[4]
grades_average = grades_total / len(grades)

print(f'Average: {grades_average:.2f}',end='\n\n')

# lists comprehension is a handy way to create lists
# taking advantage of loops, in this case I used it
# to filter the grades lower than 60.
failing_grades = [grade for grade in grades if grade < 60]

grades.remove(failing_grades[0])
grades.pop(grades.index(failing_grades[-1]))

print(f'Updated List: {grades}')

new_grades_average = sum(grades) / len(grades)

print(f'Updated Average: {new_grades_average:.3f}')

```



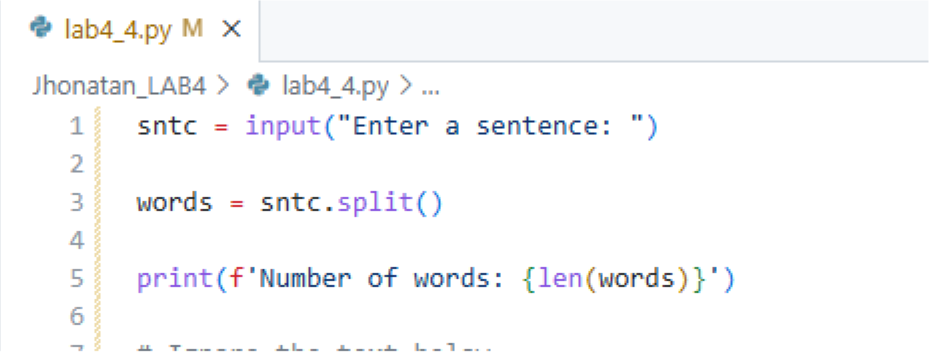
```
● @jhonatanparada499 ➜ /workspaces/Jhonatan_ET574 (main)
  Current list: [92, 51, 83, 37, 72]
  Average: 67.00

  Updated List: [92, 83, 72]
  Updated Average: 82.333
```



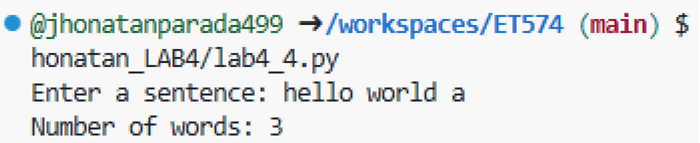
```python
courses = ['ET123','ET456','ET789','ENGL101','MA321']
print(courses)

print(f'I am taking {len(courses)} courses.')

print(courses[0],courses[-1],sep='\t')

print(courses[:4])

print(courses[-4:])

print(courses[1:-1])

```



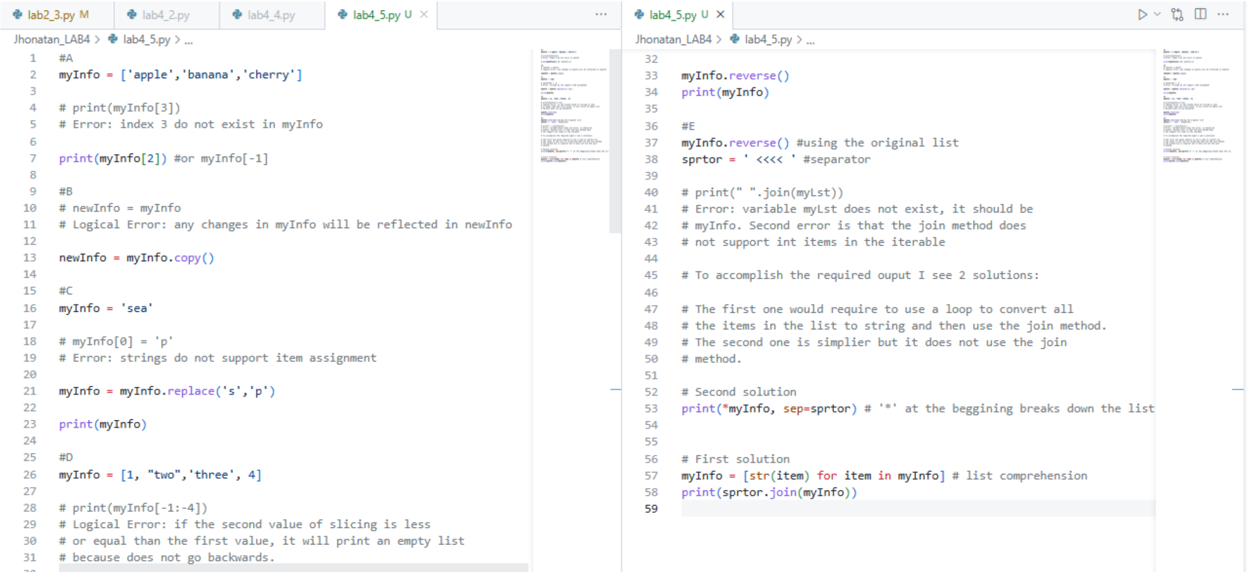
```
@jhonatanparada499 ➜/workspaces/Jhonatan_ET574 (main)
['ET123', 'ET456', 'ET789', 'ENGL101', 'MA321']
I am taking 5 courses.
ET123   MA321
['ET123', 'ET456', 'ET789', 'ENGL101']
['ET456', 'ET789', 'ENGL101', 'MA321']
['ET456', 'ET789', 'ENGL101']
```



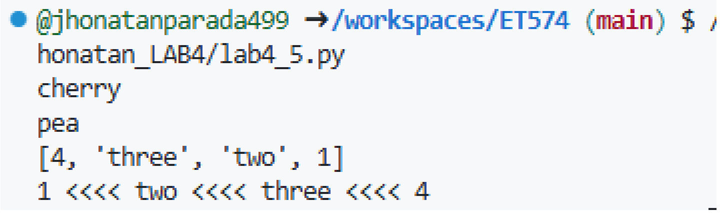
```python
sntc = input("Enter a sentence: ")

words = sntc.split()

print(f'Number of words: {len(words)}')

# Ignore the text below
```



```
@jhonatanparada499 ➜/workspaces/ET574 (main) $
honatan_LAB4/lab4_4.py
Enter a sentence: hello world a
Number of words: 3
```


lab2_3.py M
lab4_2.py
lab4_4.py
lab4_5.py U
Jhonatan_LAB4 > lab4_5.py > ...
lab4_5.py U
Jhonatan_LAB4 > lab4_5.py > ...

```python
#A
myInfo = ['apple','banana','cherry']

# print(myInfo[3])
# Error: index 3 do not exist in myInfo

print(myInfo[2]) #or myInfo[-1]

#B
# newInfo = myInfo
# Logical Error: any changes in myInfo will be reflected in newInfo

newInfo = myInfo.copy()

#C
myInfo = 'sea'

# myInfo[0] = 'p'
# Error: strings do not support item assignment

myInfo = myInfo.replace('s','p')

print(myInfo)

#D
myInfo = [1, "two",'three', 4]

# print(myInfo[-1:-4])
# Logical Error: if the second value of slicing is less
# or equal than the first value, it will print an empty list
# because does not go backwards.

myInfo.reverse()
print(myInfo)

#E
myInfo.reverse() #using the original list
sprtor = ' <<<< ' #separator

# print(" ".join(myLst))
# Error: variable myLst does not exist, it should be
# myInfo. Second error is that the join method does
# not support int items in the iterable

# To accomplish the required ouput I see 2 solutions:

# The first one would require to use a loop to convert all
# the items in the list to string and then use the join method.
# The second one is simplier but it does not use the join
# method.

# Second solution
print(*myInfo, sep=sprtor) # '*' at the beggining breaks down the list


# First solution
myInfo = [str(item) for item in myInfo] # list comprehension
print(sprtor.join(myInfo))

```



```
● @jhonatanparada499 ➜/workspaces/ET574 (main) $ /
  honatan_LAB4/lab4_5.py
  cherry
  pea
  [4, 'three', 'two', 1]
  1 <<<< two <<<< three <<<< 4
```

2. On this lesson, I learned about the string method split and join. At the beginning I tried to create a function to achieve the same result as the method split using the strip method. In the future, I want to create a way to remove the symbols such as commas and dots, that the split method gathers along with the words. That way, I would get a more accurate code in terms of logic.