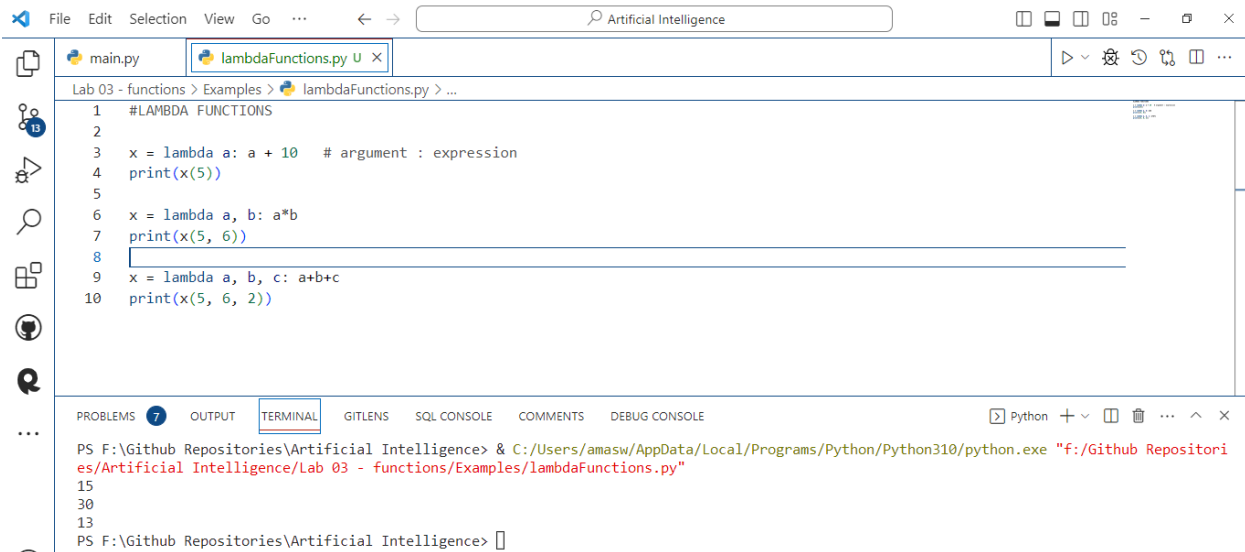


LAB 03

Examples:

Lambda Functions



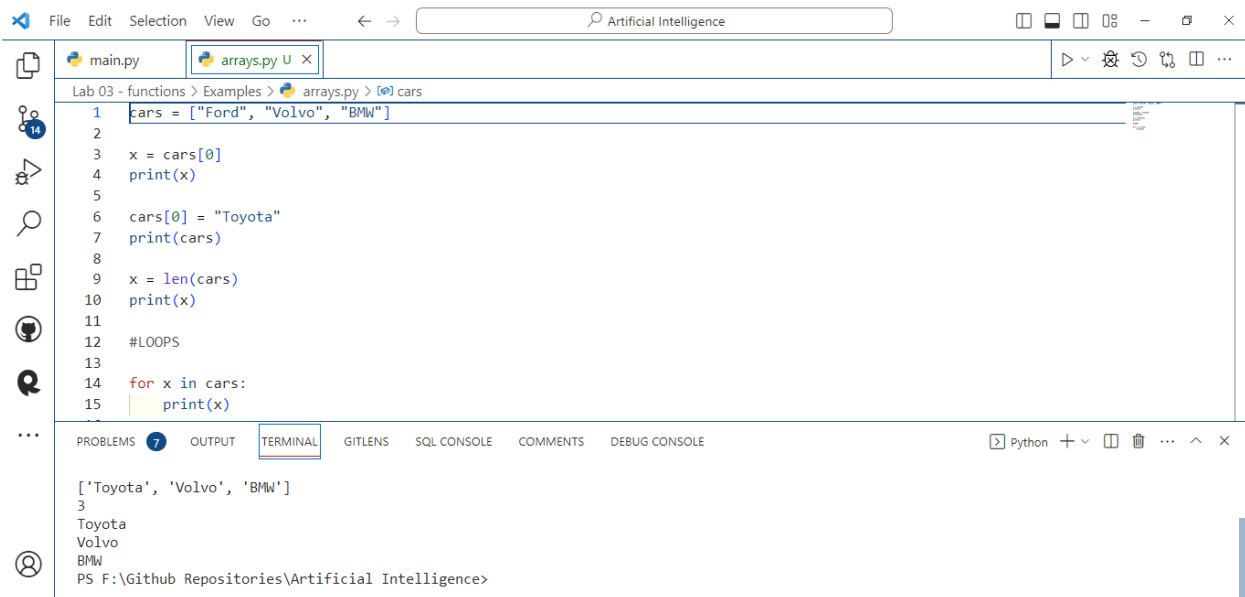
The screenshot shows the Visual Studio Code editor with a file named `lambdaFunctions.py` open. The code defines three lambda functions and prints their results. The terminal shows the command to run the script and the output.

```
1 #LAMBDA FUNCTIONS
2
3 x = lambda a: a + 10 # argument : expression
4 print(x(5))
5
6 x = lambda a, b: a*b
7 print(x(5, 6))
8
9 x = lambda a, b, c: a+b+c
10 print(x(5, 6, 2))
```

Terminal output:

```
PS F:\Github Repositories\Artificial Intelligence> & C:/Users/amasw/AppData/Local/Programs/Python/Python310/python.exe "f:\Github Repositories\Artificial Intelligence\Lab 03 - functions\Examples\lambdaFunctions.py"
15
30
13
PS F:\Github Repositories\Artificial Intelligence>
```

Arrays:



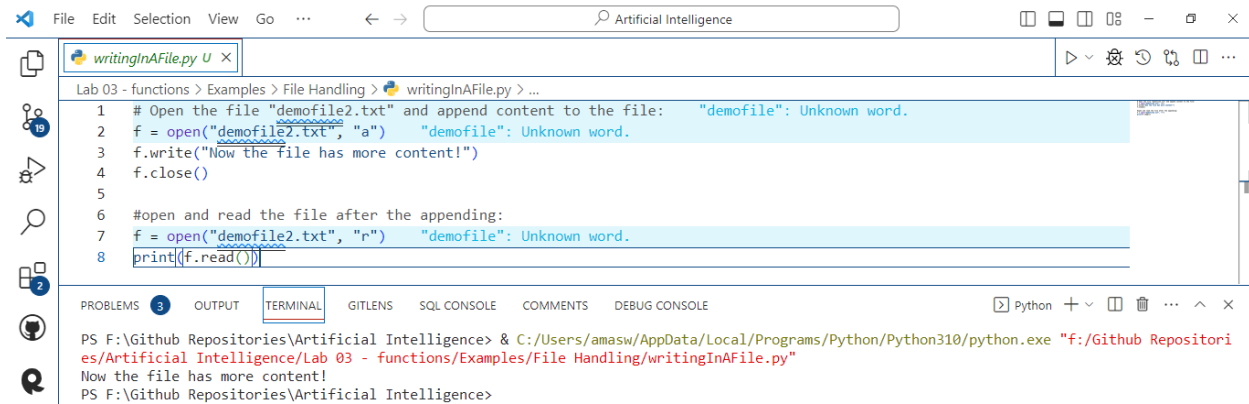
The screenshot shows the Visual Studio Code editor with a file named `arrays.py` open. The code creates a list of car names, prints the first element, changes the first element to 'Toyota', prints the list, and then prints the length of the list. The terminal shows the command to run the script and the output.

```
1 cars = ["Ford", "Volvo", "BMW"]
2
3 x = cars[0]
4 print(x)
5
6 cars[0] = "Toyota"
7 print(cars)
8
9 x = len(cars)
10 print(x)
11
12 #LOOPS
13
14 for x in cars:
15     print(x)
```

Terminal output:

```
['Toyota', 'Volvo', 'BMW']
3
Toyota
Volvo
BMW
PS F:\Github Repositories\Artificial Intelligence>
```

File Handling:

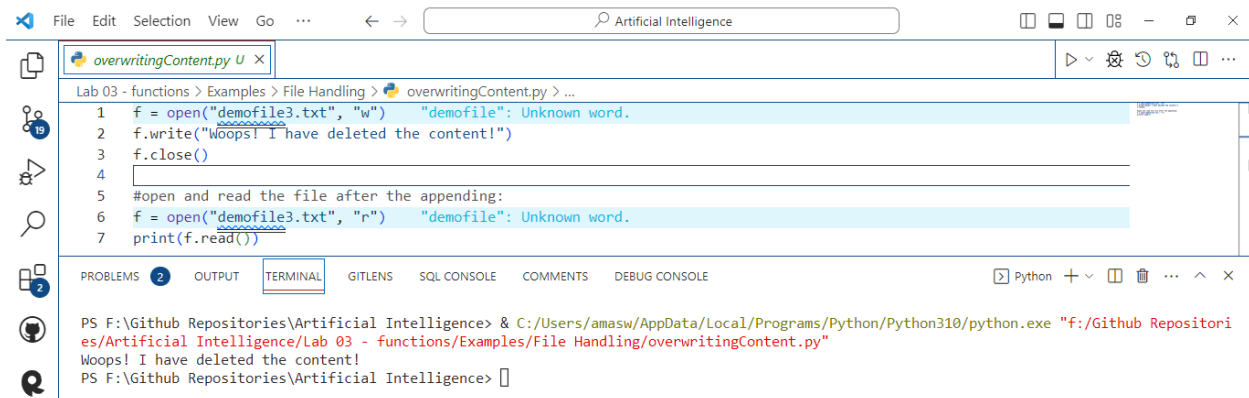


The screenshot shows the Visual Studio Code editor with a file named `writingInAFile.py` open. The file is located at `Lab 03 - functions > Examples > File Handling > writingInAFile.py`. The code in the editor is as follows:

```
1 # Open the file "demofile2.txt" and append content to the file: "demofile": Unknown word.
2 f = open("demofile2.txt", "a") "demofile": Unknown word.
3 f.write("Now the file has more content!")
4 f.close()
5
6 #open and read the file after the appending:
7 f = open("demofile2.txt", "r") "demofile": Unknown word.
8 print(f.read())
```

The terminal output shows the command to run the script and its output:

```
PS F:\Github Repositories\Artificial Intelligence> & C:/Users/amasw/AppData/Local/Programs/Python/Python310/python.exe "f:/Github Repositories/Artificial Intelligence/Lab 03 - functions/Examples/File Handling/writingInAFile.py"
Now the file has more content!
PS F:\Github Repositories\Artificial Intelligence>
```



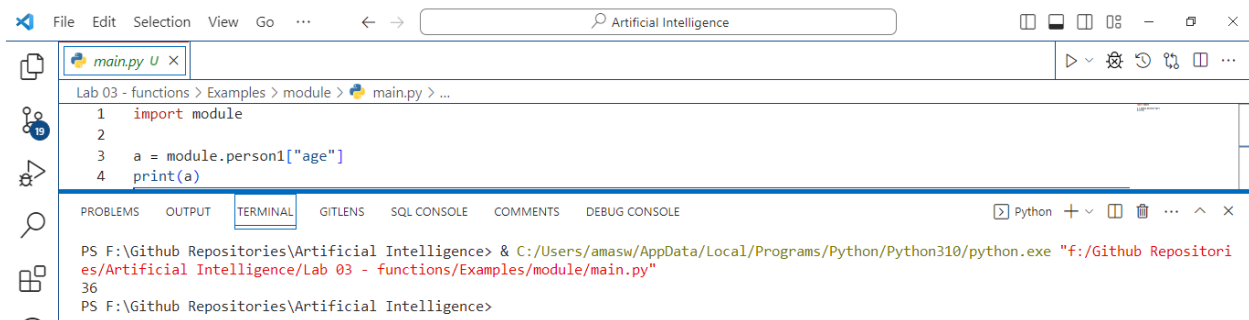
The screenshot shows the Visual Studio Code editor with a file named `overwritingContent.py` open. The file is located at `Lab 03 - functions > Examples > File Handling > overwritingContent.py`. The code in the editor is as follows:

```
1 f = open("demofile3.txt", "w") "demofile": Unknown word.
2 f.write("Woops! I have deleted the content!")
3 f.close()
4
5 #open and read the file after the appending:
6 f = open("demofile3.txt", "r") "demofile": Unknown word.
7 print(f.read())
```

The terminal output shows the command to run the script and its output:

```
PS F:\Github Repositories\Artificial Intelligence> & C:/Users/amasw/AppData/Local/Programs/Python/Python310/python.exe "f:/Github Repositories/Artificial Intelligence/Lab 03 - functions/Examples/File Handling/overwritingContent.py"
Woops! I have deleted the content!
PS F:\Github Repositories\Artificial Intelligence>
```

Module:



The screenshot shows the Visual Studio Code editor with a file named `main.py` open. The file is located at `Lab 03 - functions > Examples > module > main.py`. The code in the editor is as follows:

```
1 import module
2
3 a = module.person1["age"]
4 print(a)
```

The terminal output shows the command to run the script and its output:

```
PS F:\Github Repositories\Artificial Intelligence> & C:/Users/amasw/AppData/Local/Programs/Python/Python310/python.exe "f:/Github Repositories/Artificial Intelligence/Lab 03 - functions/Examples/module/main.py"
36
PS F:\Github Repositories\Artificial Intelligence>
```

Exercise 01:

(i). a Python program to square and cube every number in a given list of integers using Lambda.

```
arr = [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 ]
square = lambda x: x**2
cube = lambda x: x**3
print('Array: ',arr )
print('Squared Array: ',list(map(square, arr)))
print('Cubed Array: ',list(map(cube, arr)))
```

(ii). a Python program to find if a given string starts with a given character using Lambda.

```
str = input('Enter a string: ')
startVar = input('Enter a character to check if the string starts with it: ')
start = lambda x: x.startswith(startVar)
if start(str) == True:
    print(f"{str} starts with {startVar}")
else:
    print(f"{str} does not start with {startVar}")
```

(iii). a Python program to extract year, month, date and time using Lambda.

```
import datetime
currentDateTime = datetime.datetime.now()
year = lambda x: x.year
month = lambda x: x.month
day = lambda x: x.day
time = lambda x: x.time()
print('Year - ',year(currentDateTime))
print('Month - ',month(currentDateTime))
print('Day - ',day(currentDateTime))
print('Time - ',time(currentDateTime))
```

Exercise 02:

(i). You have collected information about cities in your province. You decide to store each city's name, population, and mayor in a file. Write a python program to accept the data for a number of cities from the keyboard and store the data in a file in the order in which they're entered.

```
# opening a file
with open("cityData.txt", "w") as f:

    # getting the number of cities
    citiesNum = int(input("Enter umber of cities: "))

    # looping through the number of cities
    for i in range(citiesNum):
        cityName = input("Enter city name: ")
        cityPopulation = input("Enter city population: ")
        cityMayor = input("Enter city mayor: ")

        # Writing the message to the file
        f.write(f"City Name: {cityName} City Population: {cityPopulation} City
Mayor: {cityMayor} \n")
```

(ii). Write a python program to create a data file student.txt and append the message "Now we are AI students's"

```
# opening a file
with open("student.txt", "a") as f:

    # Writing the message to the file
    f.write("Now we are AI students")
```