Course Number and Name: CSE 4618 Artificial Intelligence Lab	
Report: Lab 0 - Python/Autograder Tutorial	
Student Name: Nafisa Maliyat	**Student ID:** 200042133
Date Of Submission: 20 January, 2024	**Submitted to:** Md. Bakhtiar Hasan, Assistant Professor, CSE

Overview:

This lab provided an introduction to Python and Autograder. There was three tasks provided as well as tutorial.zip file. After solving each task, Autograder was used to evaluate whether or not it was the correct solution.

On the next pages, I have mentioned the following :

- the problem statement
- analysis of the problem

addition.py:

Problem Statement:

Implementation of a simple function that takes two parameters and return their addition.

Analysis of the problem:

This required simply using '+' operator to sum the two parameters a and b.

```
return a+b
```

buyLotsOfFruit.py:

Problem Statement:

Calculate the total cost of buying some fruits from a predefined dictionary fruitPrices. The case where the fruit does not exist in the dictionary must be handled as well.

Analysis of the problem:

The function takes the order list containing the fruits purchased. A variable initial_cost is initialized to 0 to count the total cost. Each item in the list is iterated and the fruit is searched in the fruitPrices.

```
    totalCost = 0.0
    for fruit, pounds in orderList:
```

If the fruitPrices does not contain the fruit, a warning message is printed and None is returned instead of the cost.

```
            print(f"{fruit} is not present in fruitPrices!")
            return None
```

Otherwise, the unit price of the fruit price is multiplied with the quantity of the fruit purchased.

```
        if fruit in fruitPrices:
            totalCost+= pounds * fruitPrices[fruit]
```

The total cost is returned once all the fruits have been iterated through.

Result:

```
(cse4618) PS D:\6th Semester\CSE 4618 Artificial Intelligence Lab\CSE-4618-Artificial-Intelligence>p
ython Lab0/buyLotsOfFruit.py
Cost of [('apples', 2.0), ('pears', 3.0), ('limes', 4.0)] is 12.25
```

shopSmart.py:

Problem Statement:

Implementation of shopSmart function that takes in a list of orders containing the fruit name and quantity and list of shops fruitShops. The task requires calculation of total cost of the whole order list for each shop and return the shop that would give the lowest total cost.

Analysis of the problem:

Two variables are used to keep track of the minimum cost and the shop containing the minimum cost for comparison. The variable keeping track of minimum cost and the shop are initialized.

```
    minimumCost = float('inf')
    shopWithMinimumCost = None
```

The shop list is iterated for each shop and the total cost is obtained using the shop's predefined function getPriceOfOrder in shop.py function which takes in the order list and returns the total cost according to the shop's pricing list.

```
  for fruitShop in fruitShops:
       costForThisShop = float(fruitShop.getPriceOfOrder(orderList))
```

If any shop returns a total cost of less than the minimum cost, the variables storing minimum cost and the corresponding shop are updated.

```
       if(costForThisShop<minimumCost):
           shopWithMinimumCost=fruitShop
           minimumCost=costForThisShop
```

At the end of the loop, the shop name stored in the variable shopWithMinimumCost is

returned as the result.

Result: