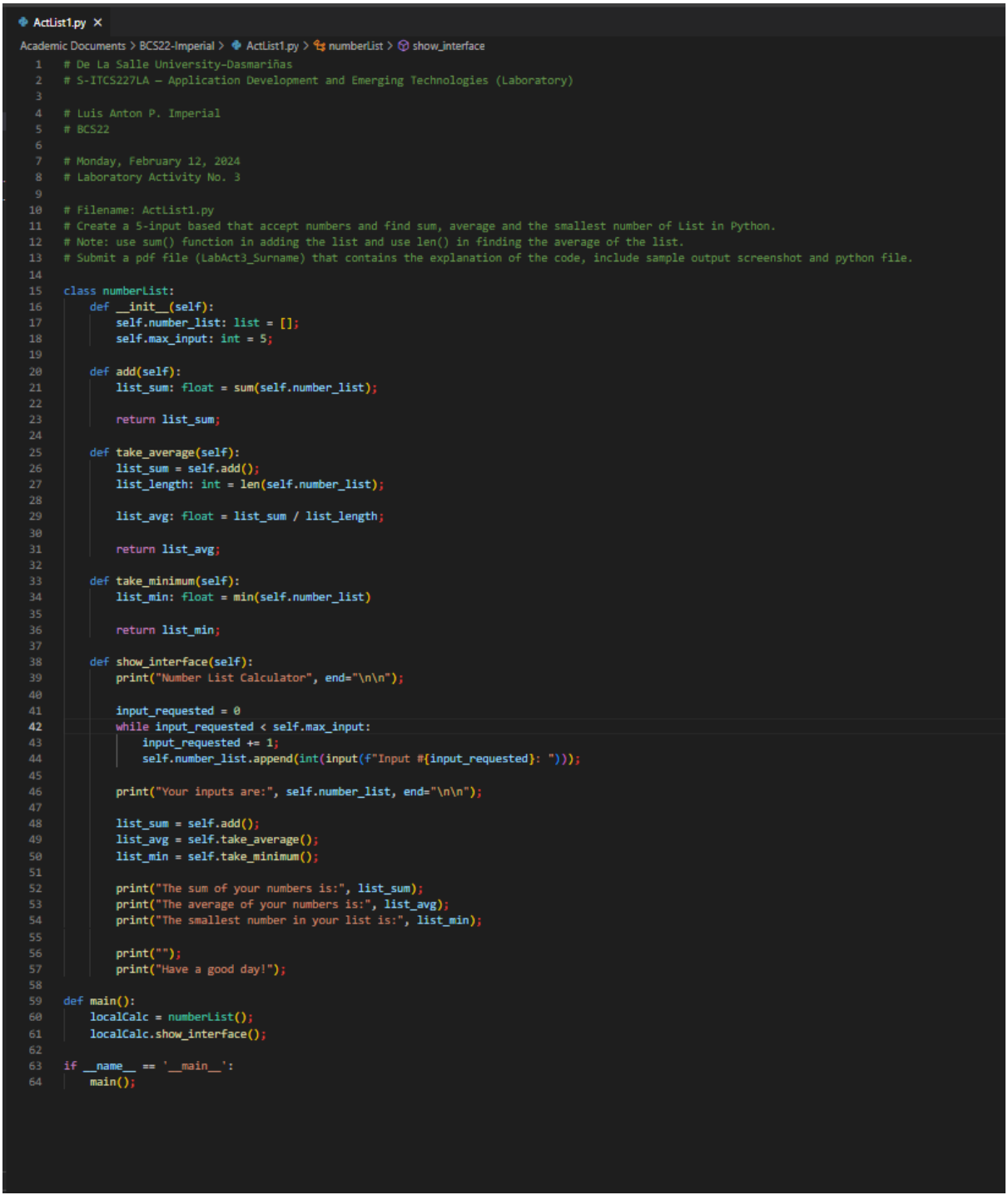
For our sixth class in Application Development and Emerging Technologies, we made a script showing the sum, average and minimum element in a list consisting of 5 numerical inputs.

Code



ActList1.py

Academic Documents > BCS22-Imperial > ActList1.py > numberList > show_interface

```python
# De La Salle University-Dasmariñas
# S-ITCS227LA – Application Development and Emerging Technologies (Laboratory)

# Luis Anton P. Imperial
# BCS22

# Monday, February 12, 2024
# Laboratory Activity No. 3

# Filename: ActList1.py
# Create a 5-input based that accept numbers and find sum, average and the smallest number of List in Python.
# Note: use sum() function in adding the list and use len() in finding the average of the list.
# Submit a pdf file (LabAct3_Surname) that contains the explanation of the code, include sample output screenshot and python file.

class numberList:
    def __init__(self):
        self.number_list: list = [];
        self.max_input: int = 5;

    def add(self):
        list_sum: float = sum(self.number_list);

        return list_sum;

    def take_average(self):
        list_sum = self.add();
        list_length: int = len(self.number_list);

        list_avg: float = list_sum / list_length;

        return list_avg;

    def take_minimum(self):
        list_min: float = min(self.number_list)

        return list_min;

    def show_interface(self):
        print("Number List Calculator", end="\n\n");

        input_requested = 0
        while input_requested < self.max_input:
            input_requested += 1;
            self.number_list.append(int(input(f"Input #{input_requested}: ")));

        print("Your inputs are:", self.number_list, end="\n\n");

        list_sum = self.add();
        list_avg = self.take_average();
        list_min = self.take_minimum();

        print("The sum of your numbers is:", list_sum);
        print("The average of your numbers is:", list_avg);
        print("The smallest number in your list is:", list_min);

        print("");
        print("Have a good day!");

def main():
    localCalc = numberList();
    localCalc.show_interface();

if __name__ == '__main__':
    main();
```

Explanation

Code	Description
# De La Salle University–Dasmariñas # S-ITCS227LA — Application Development and Emerging Technologies (Laboratory) # Luis Anton P. Imperial # BCS22 # Monday, February 12, 2024 # Laboratory Activity No. 3 # Filename: ActList1.py # Create a 5-input based that accept numbers and find sum, average and the smallest number of List in Python. # Note: use sum() function in adding the list and use len() in finding the average of the list. # Submit a pdf file (LabAct3_Surname) that contains the explanation of the code, include sample output screenshot and python file.	Comments, describing the script overall. These are usually placed in the header of the source code, detailing authorship and the organization encompassing said author(s).

class numberList:
  def __init__(self):
  self.number_list: list = [];
  self.max_input: int = 5;
	Initiate a new class, which will contain all the functions we need to collect inputs and compute. The number list is, for now, empty. The maximum input, as given, is 5.

def add(self):
  list_sum: float = sum(self.number_list);
 	
  return list_sum;
	Using the `sum()` function, we are performing the 'addition' operation on all the components of the list, and returning it to be placed in a variable.

def take_average(self):
  list_sum = self.add();
  list_length: int = len(self.number_list);
 	
  list_avg: float = list_sum / list_length;
 	
  return list_avg;
	Taking the average means getting the sum of a set of numbers, and dividing it by the "length" of the set. In the programming space, the length of a list is how many elements it contains.

def take_minimum(self): list_min: float = min(self.number_list)	This does the same as our `add()` function, but for taking the minimum value available in the list.

return list_min;	

def show_interface(self):
print("Number List Calculator", end="\n\n");
 	
input_requested = 0
while input_requested < self.max_input:
input_requested += 1;
 	
self.number_list.append(int(input(f"Input #{input_requested}: ")));
	Let's create a function which will collect the input, and display the output, in a Terminal User Interface (TUI).
	
	Firstly, we need the inputs, so we will repeatedly ask for them until the requests reach the maximum specified at the start of the class.

print("Your inputs are:", self.number_list, end="\n\n");	Then, we display those inputs.

list_sum = self.add();
list_avg = self.take_average();
list_min = self.take_minimum();
	We use the functions we created earlier, in order to prepare for displaying them to the end-user.

print("The sum of your numbers is:", list_sum); print("The average of your numbers is:", list_avg); print("The smallest number in your list is:", list_min);	And now, we display them!
print(""); print("Have a good day!");	Buh-bye!

def main():
localCalc = numberList();
localCalc.show_interface();
	Of course, a class won't run on its own, which is why we should instantiate an object defined as the class we made.

if __name__ == '__main__': main();	After creating a function for said object, let's run it!

Output

```
PS D:\Shared Documents> py "Academic Documents/BCS22-Imperial/ActList1.py"
Number List Calculator

Input #1: 5
Input #2: 10
Input #3: 15
Input #4: 20
Input #5: 25
Your inputs are: [5, 10, 15, 20, 25]

The sum of your numbers is: 75
The average of your numbers is: 15.0
The smallest number in your list is: 5
Have a good day!
```

Instructions

<table>
<tr><th colspan="2">Laboratory Activity No. 3</th></tr>
<tr><td rowspan="6">Filename: ActList1.py

Create a 5-input based that accept numbers and find sum, average and the smallest number of List in Python.

Note: use sum() function in adding the list and use len() in finding the average of the list.

Submit a pdf file (LabAct3_Surname) that contains the explanation of the code, include sample output screenshot and python file.</td><td>Type: Dropbox</td></tr>
<tr><td>Max score: 40</td></tr>
<tr><td>Category: Enabling Asssessment</td></tr>
<tr><td>Start: Feb 12, 10:00 am</td></tr>
<tr><td>Due: Feb 12, 1:00 pm</td></tr>
<tr><td>Max. attempts: 2</td></tr>
<tr><td>Allow late submissions: No</td></tr>
</table>