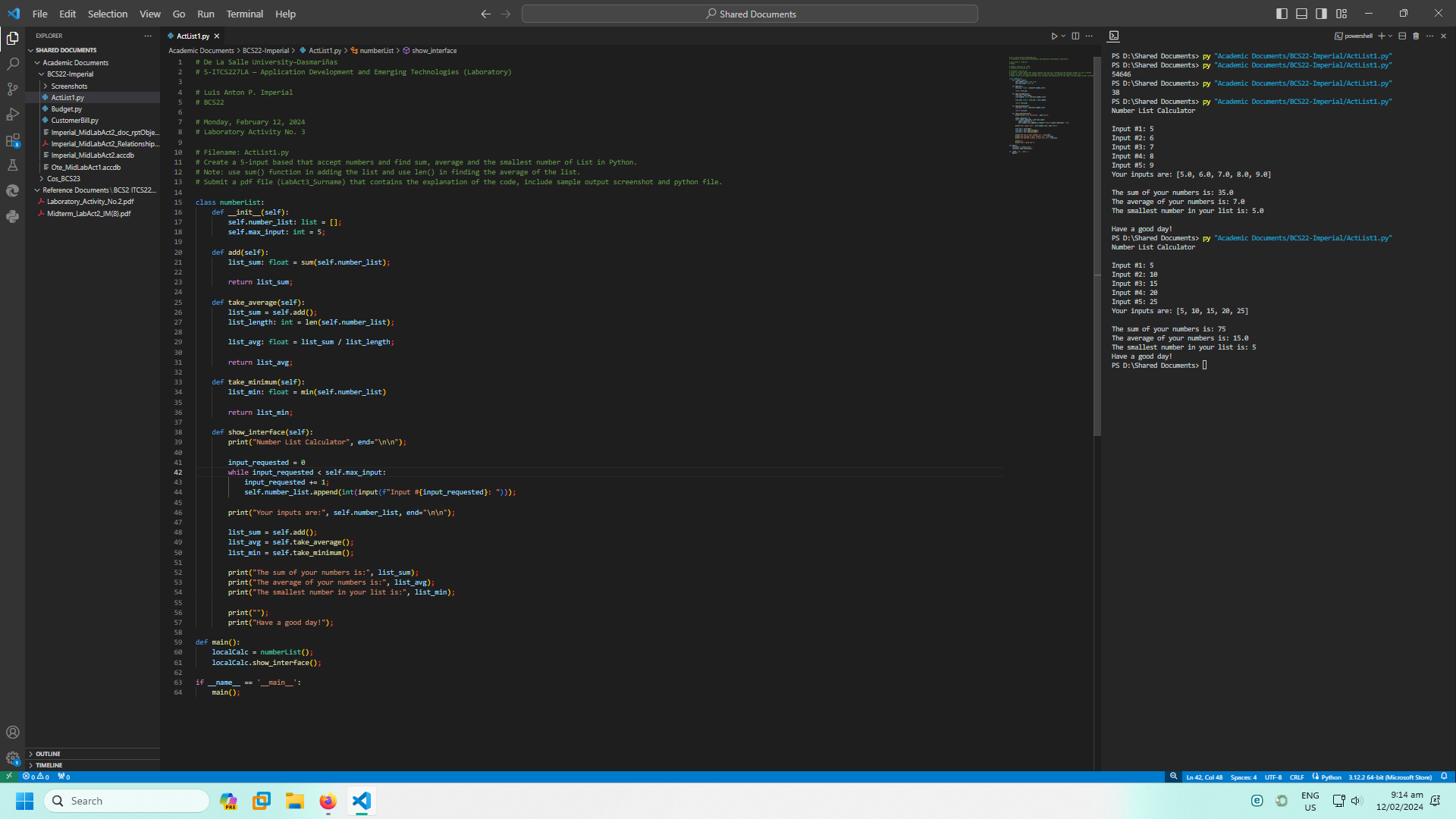
For our sixth class in Application Development and Emerging Technologies, we made a script showing the sum, average and minimum element in a list consisting of 5 numerical inputs.

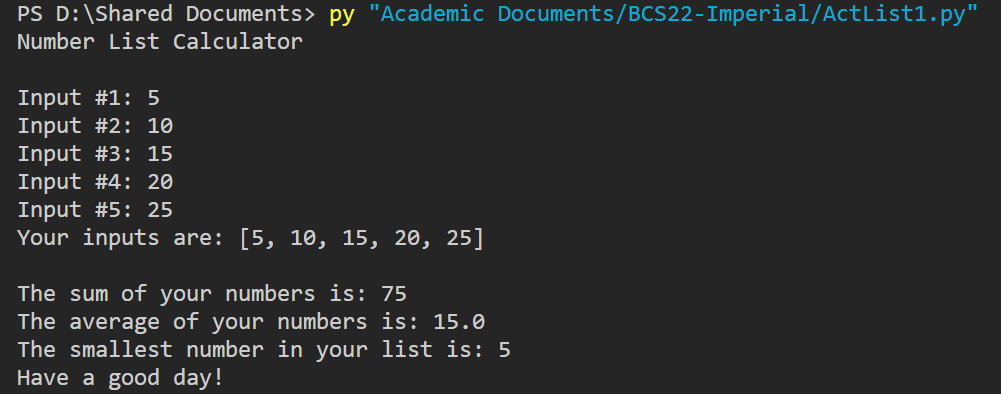
# Code



## 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)    return list\_min; | This does the same as our `add()` function, but for taking the minimum value available in the list. |
| 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



# Instructions

|  |  |
| --- | --- |
| 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. | **Type:** Dropbox |
| **Max score:** 40 |
| **Category:** Enabling Asssessment |
| **Start:** Feb 12, 10:00 am |
| **Due:** Feb 12, 1:00 pm |
| **Max. attempts:** 2 |
| **Allow late submissions:** No |