

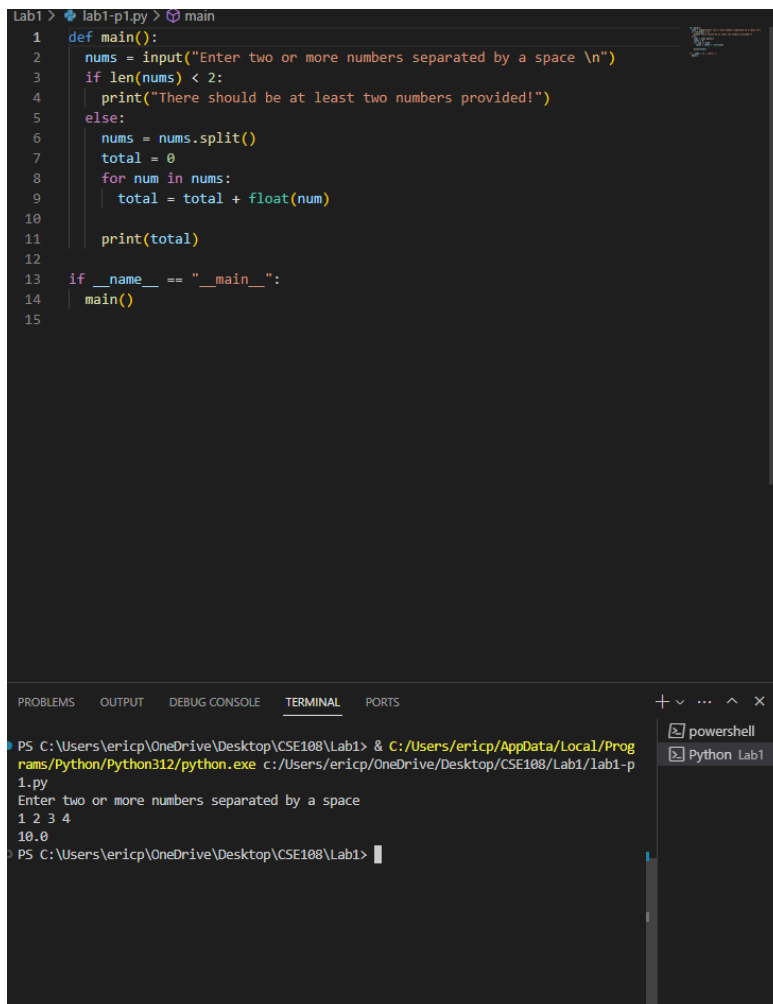
Eric Ponce

1/31/24

Lab1 Report

Problem #1:

We were asked to create a program that adds numbers together. The numbers are determined by the user in a space separated list. Decimal values should also be valid.



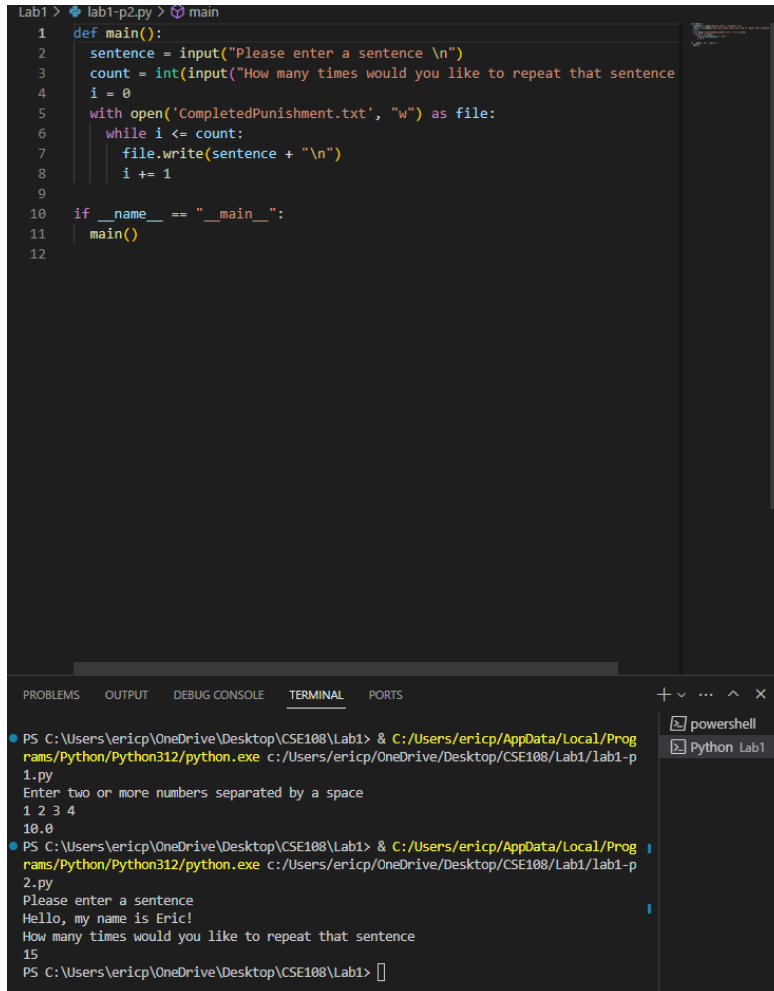
```
Lab1 > lab1-p1.py > main
1  def main():
2      nums = input("Enter two or more numbers separated by a space \n")
3      if len(nums) < 2:
4          print("There should be at least two numbers provided!")
5      else:
6          nums = nums.split()
7          total = 0
8          for num in nums:
9              total = total + float(num)
10
11         print(total)
12
13 if __name__ == "__main__":
14     main()
15
```

```
PS C:\Users\ericp\OneDrive\Desktop\CSE108\Lab1> & C:/Users/ericp/AppData/Local/Programs/Python/Python312/python.exe c:/Users/ericp/OneDrive/Desktop/CSE108/Lab1/lab1-p1.py
Enter two or more numbers separated by a space
1 2 3 4
10.0
PS C:\Users\ericp\OneDrive\Desktop\CSE108\Lab1>
```

Here my program starts off by asking the user to input numbers separated by a space. If the user enters less than 2 numbers, an error is thrown saying there should be at least 2 numbers entered. The numbers are technically one giant string so we must first separate the numbers into separate strings and convert them to floats. They are all added together, and the sum is returned to the user.

Problem #2:

We were asked to write a program that asks the user to type a sentence and the number of times they want the sentence repeated.



The image shows a Visual Studio Code editor window with a Python script named `lab1-p2.py` and a terminal window below it. The script defines a `main` function that prompts the user for a sentence and a count, then writes the sentence to a file named `CompletedPunishment.txt` the specified number of times. The terminal shows the execution of the script, with the user entering the sentence "Hello, my name is Eric!" and the count "15".

```
Lab1 > lab1-p2.py main
1 def main():
2     sentence = input("Please enter a sentence \n")
3     count = int(input("How many times would you like to repeat that sentence
4     i = 0
5     with open('CompletedPunishment.txt', "w") as file:
6         while i <= count:
7             file.write(sentence + "\n")
8             i += 1
9
10 if __name__ == "__main__":
11     main()
12
```

Terminal Output:

```
PS C:\Users\ericp\OneDrive\Desktop\CSE108\Lab1> & C:/Users/ericp/AppData/Local/Programs/Python/Python312/python.exe c:/Users/ericp/OneDrive/Desktop/CSE108/Lab1/lab1-p1.py
Enter two or more numbers separated by a space
1 2 3 4
10.0
PS C:\Users\ericp\OneDrive\Desktop\CSE108\Lab1> & C:/Users/ericp/AppData/Local/Programs/Python/Python312/python.exe c:/Users/ericp/OneDrive/Desktop/CSE108/Lab1/lab1-p2.py
Please enter a sentence
Hello, my name is Eric!
How many times would you like to repeat that sentence
15
PS C:\Users\ericp\OneDrive\Desktop\CSE108\Lab1>
```

```
Lab1 > CompletedPunishment.txt
1 Hello, my name is Eric!
2 Hello, my name is Eric!
3 Hello, my name is Eric!
4 Hello, my name is Eric!
5 Hello, my name is Eric!
6 Hello, my name is Eric!
7 Hello, my name is Eric!
8 Hello, my name is Eric!
9 Hello, my name is Eric!
10 Hello, my name is Eric!
11 Hello, my name is Eric!
12 Hello, my name is Eric!
13 Hello, my name is Eric!
14 Hello, my name is Eric!
15 Hello, my name is Eric!
16

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\erick\OneDrive\Desktop\CSE108\Lab1> & C:/Users/erick/AppData/Local/Programs/Python/Python312/python.exe c:/Users/erick/OneDrive/Desktop/CSE108/Lab1/lab1-p
2.py
Please enter a sentence
Hello, my name is Eric!
How many times would you like to repeat that sentence
15
PS C:\Users\erick\OneDrive\Desktop\CSE108\Lab1> & C:/Users/erick/AppData/Local/Programs/Python/Python312/python.exe c:/Users/erick/OneDrive/Desktop/CSE108/Lab1/lab1-p
2.py
Please enter a sentence
Hello, my name is Eric!
How many times would you like to repeat that sentence
15
PS C:\Users\erick\OneDrive\Desktop\CSE108\Lab1>
```

Here my program is asking the user to enter a sentence which is being stored in the sentence variable and they are also being asked to enter the number of times they want that sentence repeated. That value is being stored in the count variable. The program then opens the CompletedPunishment.txt file so that we can write data in it. A while loops is completed a user defined number of times based off of the number they previously entered. Each iteration of the loop writes the user entered sentence to the file.

Problem #3:

We were asked to write a word count program that counts the number of times a user entered word comes up in a predetermined text file.

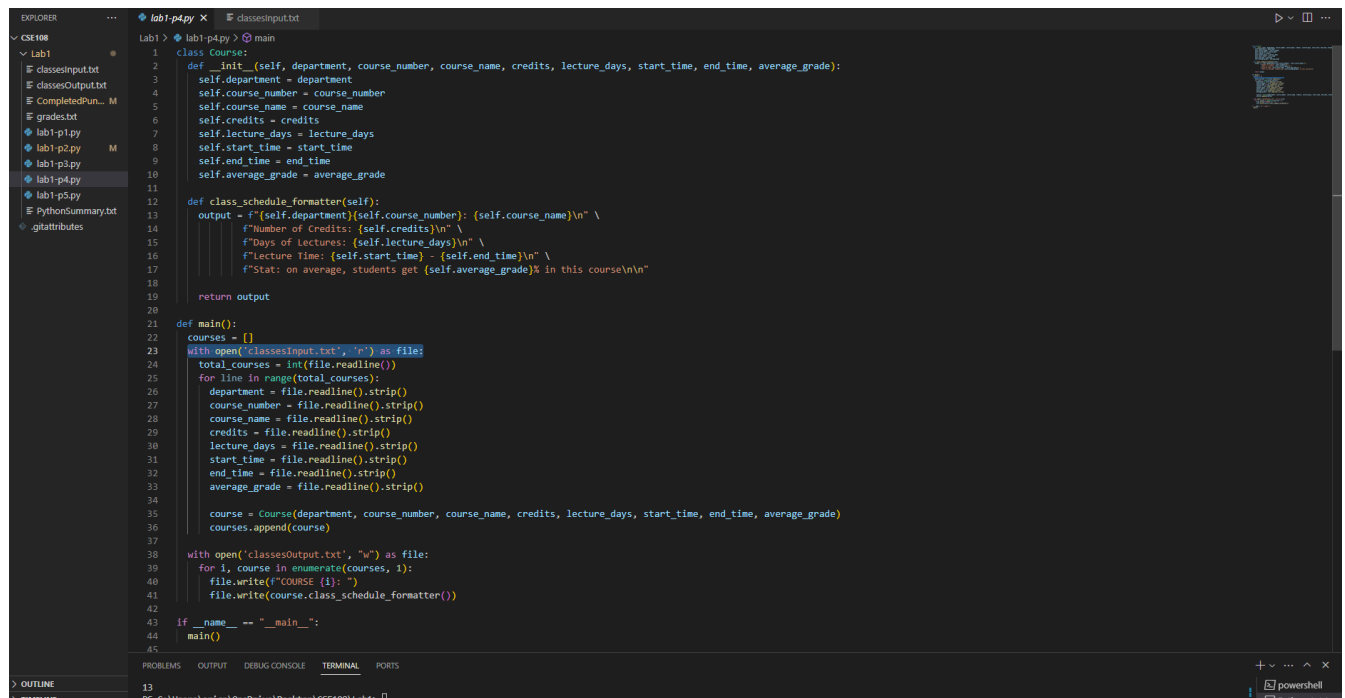
```
1  def main():
2      word = input("Please enter a word \n")
3      total = 0
4
5      with open('PythonSummary.txt', "r") as file:
6          for line in file:
7              line = line.lower().replace('.', ' ').replace('-', ' ')
8              words = line.split()
9              total += words.count(word)
10
11         print(total)
12
13  if __name__ == "__main__":
14      main()
```

15
PS C:\Users\ericp\OneDrive\Desktop\CSE108\Lab1> & C:/Users/ericp/AppData/Local/Programs/Python/Python312/python.exe c:/Users/ericp/OneDrive/Desktop/CSE108/Lab1/lab1-p2.py
● Please enter a sentence
Hello, my name is Eric!
How many times would you like to repeat that sentence
15
PS C:\Users\ericp\OneDrive\Desktop\CSE108\Lab1> & C:/Users/ericp/AppData/Local/Programs/Python/Python312/python.exe c:/Users/ericp/OneDrive/Desktop/CSE108/Lab1/lab1-p3.py
● 3.py
Please enter a word
python
13
PS C:\Users\ericp\OneDrive\Desktop\CSE108\Lab1>

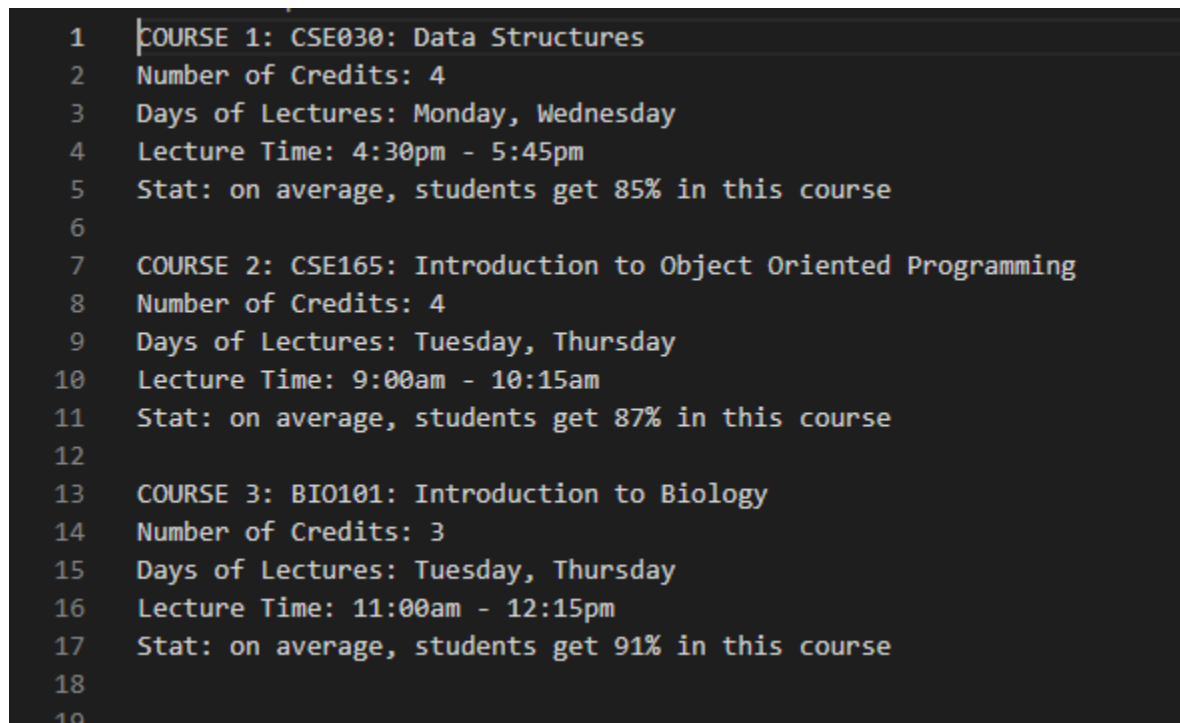
Here my program starts out by asking the user to input a word they wish to search for. The program then opens the PythonSummary.txt file and goes through each line and makes everything lowercase. It also replaces and punctuation with a blank space. The lines are split so that we are left with just words. The user entered word is compared to all the words from the text file and each time a match comes up, the total variable is increased by one. The total is then printed to the user.

Problem #4:

We were asked to view the information in the classesInput.txt file and automatically format the data in a specified way.



```
1 class Course:
2     def __init__(self, department, course_number, course_name, credits, lecture_days, start_time, end_time, average_grade):
3         self.department = department
4         self.course_number = course_number
5         self.course_name = course_name
6         self.credits = credits
7         self.lecture_days = lecture_days
8         self.start_time = start_time
9         self.end_time = end_time
10        self.average_grade = average_grade
11
12    def class_schedule_formatter(self):
13        output = f"(self.department){self.course_number}: {self.course_name}\n" \
14                f"Number of Credits: {self.credits}\n" \
15                f"Days of Lectures: {self.lecture_days}\n" \
16                f"Lecture Time: {self.start_time} - {self.end_time}\n" \
17                f"Stat: on average, students get {self.average_grade}% in this course\n\n"
18
19        return output
20
21 def main():
22     courses = []
23     with open('classesInput.txt', 'r') as file:
24         total_courses = int(file.readline())
25         for line in range(total_courses):
26             department = file.readline().strip()
27             course_number = file.readline().strip()
28             course_name = file.readline().strip()
29             credits = file.readline().strip()
30             lecture_days = file.readline().strip()
31             start_time = file.readline().strip()
32             end_time = file.readline().strip()
33             average_grade = file.readline().strip()
34
35             course = Course(department, course_number, course_name, credits, lecture_days, start_time, end_time, average_grade)
36             courses.append(course)
37
38     with open('classesOutput.txt', 'w') as file:
39         for i, course in enumerate(courses, 1):
40             file.write(f"COURSE {i}: ")
41             file.write(course.class_schedule_formatter())
42
43 if __name__ == "__main__":
44     main()
45
```



```
1 COURSE 1: CSE030: Data Structures
2 Number of Credits: 4
3 Days of Lectures: Monday, Wednesday
4 Lecture Time: 4:30pm - 5:45pm
5 Stat: on average, students get 85% in this course
6
7 COURSE 2: CSE165: Introduction to Object Oriented Programming
8 Number of Credits: 4
9 Days of Lectures: Tuesday, Thursday
10 Lecture Time: 9:00am - 10:15am
11 Stat: on average, students get 87% in this course
12
13 COURSE 3: BIO101: Introduction to Biology
14 Number of Credits: 3
15 Days of Lectures: Tuesday, Thursday
16 Lecture Time: 11:00am - 12:15pm
17 Stat: on average, students get 91% in this course
18
19
```

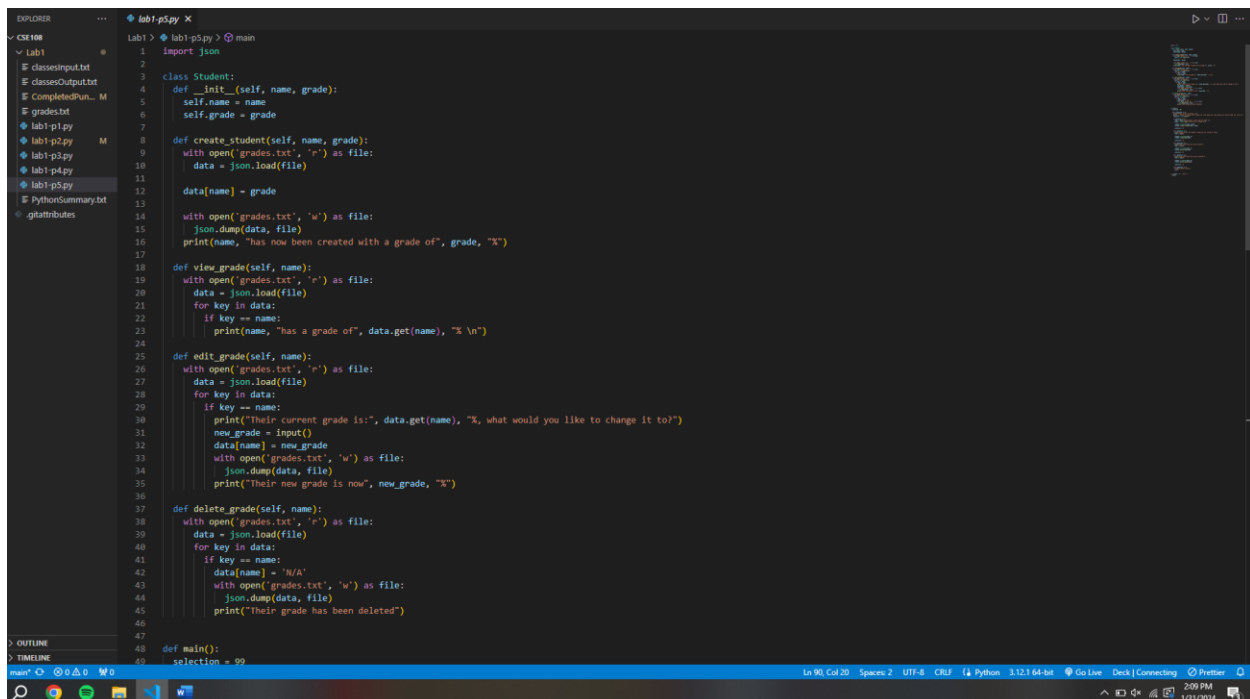
Here my program has a Course class which houses a class_schedule_formatter function that automatically formats the class data from the classesInput.txt file. We start out by opening the classesInput.txt file and reading the first line which tells us how many courses there are. Each course's data takes up 8 lines in the text file. Each of the 8 lines is read and stored in a corresponding variable. These variables are then passed to the Course class. We now open the file, but this time we plan on

writing to the file. The program calls the class `_schedule_formatter` which takes all of the variables within the course object and formats them in a specified way according to the lab instructions.

Problem #5:

We were asked to write a program that allows for the following features:

- Create a new student with a grade.
- View a student's grade when searched by full name.
- Edit a student's grade.
- Delete a student's grade.



```
1 import json
2
3 class Student:
4     def __init__(self, name, grade):
5         self.name = name
6         self.grade = grade
7
8     def create_student(self, name, grade):
9         with open('grades.txt', 'r') as file:
10             data = json.load(file)
11         data[name] = grade
12
13         with open('grades.txt', 'w') as file:
14             json.dump(data, file)
15
16         print(name, "has now been created with a grade of", grade, "X")
17
18     def view_grade(self, name):
19         with open('grades.txt', 'r') as file:
20             data = json.load(file)
21         for key in data:
22             if key == name:
23                 print(name, "has a grade of", data.get(name), "X\n")
24
25     def edit_grade(self, name):
26         with open('grades.txt', 'r') as file:
27             data = json.load(file)
28         for key in data:
29             if key == name:
30                 print("their current grade is:", data.get(name), "X, what would you like to change it to?")
31                 new_grade = input()
32                 data[name] = new_grade
33                 with open('grades.txt', 'w') as file:
34                     json.dump(data, file)
35                 print("their new grade is now", new_grade, "X")
36
37     def delete_grade(self, name):
38         with open('grades.txt', 'r') as file:
39             data = json.load(file)
40         for key in data:
41             if key == name:
42                 data[name] = 'N/A'
43                 with open('grades.txt', 'w') as file:
44                     json.dump(data, file)
45                 print("their grade has been deleted")
46
47
48 def main():
49     selection = 99
```

The screenshot shows a VS Code editor with a file explorer on the left. The file explorer shows a project named 'lab1' with several files: 'classesInput.txt', 'classesOutput.txt', 'CompletedPun...', 'grades.txt', 'lab1-pl1.py', 'lab1-pl2.py', 'lab1-pl3.py', 'lab1-pl4.py', 'lab1-pl5.py', 'PythonSummary.txt', and '.gitattributes'. The main editor window shows the 'view_grade' method of a 'Student' class. The code is as follows:

```
def main():
    selection = 99

    while selection != 5:
        print("Please select an option: \n")
        print("1. Create a student with a grade \n2. View grade \n3. Edit grade \n4. Delete grade \n5. Exit \n")
        selection = int(input())

        if selection == 1:
            name = input("Please enter a first and last name \n")
            grade = float(input("Please enter a grade \n"))

            student = Student(name, grade)
            student.create_student(name, grade)

            selection = 0

        elif selection == 2:
            print("Please enter the student's grade you are trying to view")
            name = input()

            student = Student(name, 0)
            student.view_grade(name)

            selection = 0

        elif selection == 3:
            print("Whos grade would you like to edit?")
            name = input()

            student = Student(name, 0)
            student.edit_grade(name)

            selection = 0

        elif selection == 4:
            print("Whos grade would you like to delete?")
            name = input()

            student = Student(name, 0)
            student.delete_grade(name)

            selection = 0

        elif selection == 5:
            print("Program closed!")
            break
```

The screenshot shows a VS Code editor with a file explorer on the left. The file explorer shows a project named 'lab1' with several files: 'classesInput.txt', 'classesOutput.txt', 'CompletedPun...', 'grades.txt', 'lab1-pl1.py', 'lab1-pl2.py', 'lab1-pl3.py', 'lab1-pl4.py', 'lab1-pl5.py', 'PythonSummary.txt', and '.gitattributes'. The main editor window shows the 'main' function of a 'Student' class. The code is as follows:

```
def main():
    selection = 0

    elif selection == 2:
        print("Please enter the student's grade you are trying to view")
        name = input()

        student = Student(name, 0)
        student.view_grade(name)

        selection = 0

    elif selection == 3:
        print("Whos grade would you like to edit?")
        name = input()

        student = Student(name, 0)
        student.edit_grade(name)

        selection = 0

    elif selection == 4:
        print("Whos grade would you like to delete?")
        name = input()

        student = Student(name, 0)
        student.delete_grade(name)

        selection = 0

    elif selection == 5:
        print("Program closed!")
        break

    if __name__ == "__main__":
        main()
```

Here, my program is running on an infinite loop that outputs some options to the user based on the required features. If 1 is entered, the user can create a student. If 2 is entered, the user can look up a student's grade. If 3 is entered, the user can edit a student's grade. If 4 is entered, the user can delete a student's grade. And lastly, if 5 is entered, the program ends. A student class has been created that contains various functions that assist with the required features. There are create_student, view_grade, edit_grade, and delete_grade functions. Each function does exactly what the name implies. They all

usually start out by opening the grades.txt file and reading the data within it. The data is loaded to the python program as a dictionary and edited according to the specific needs.