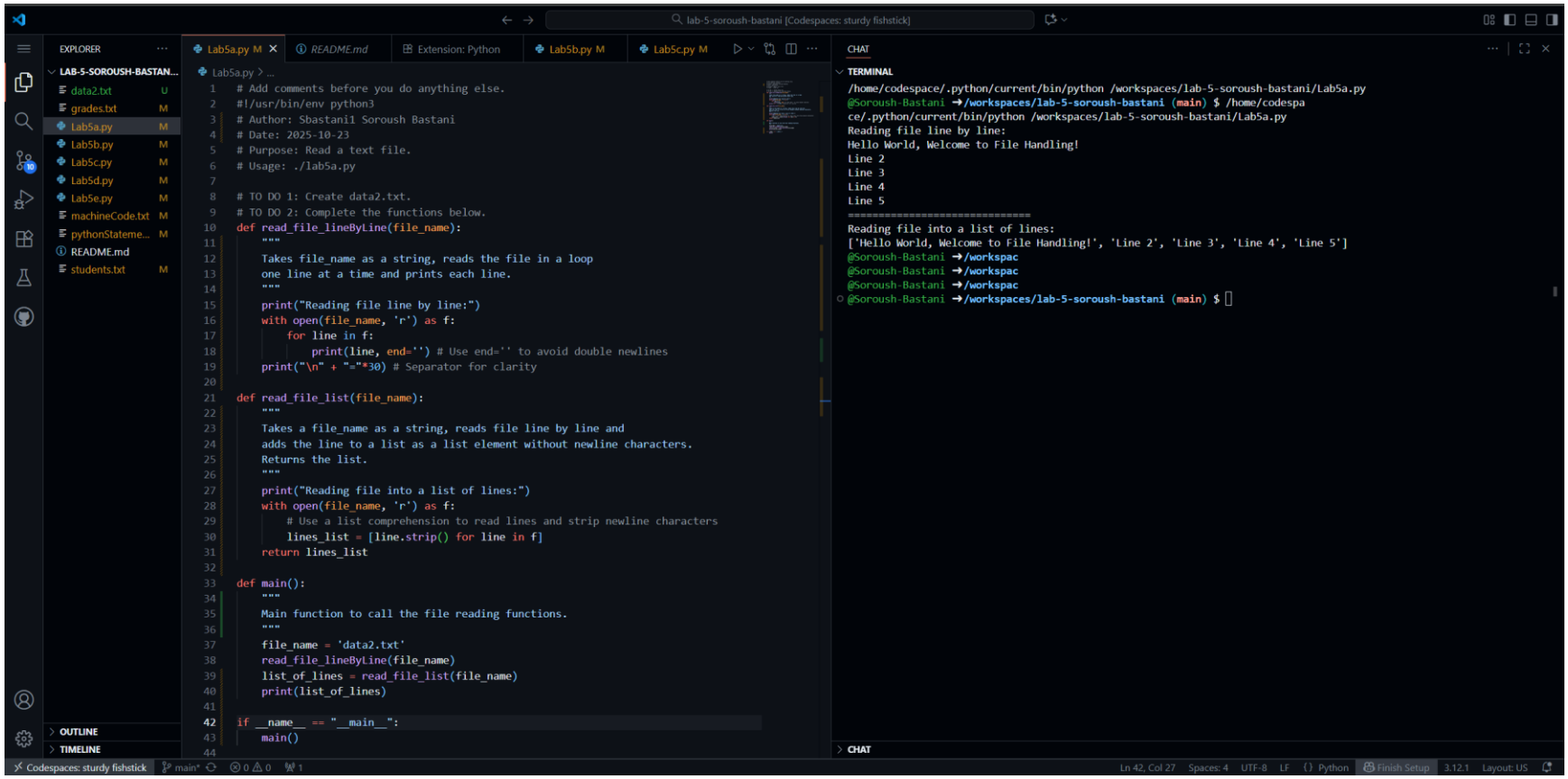
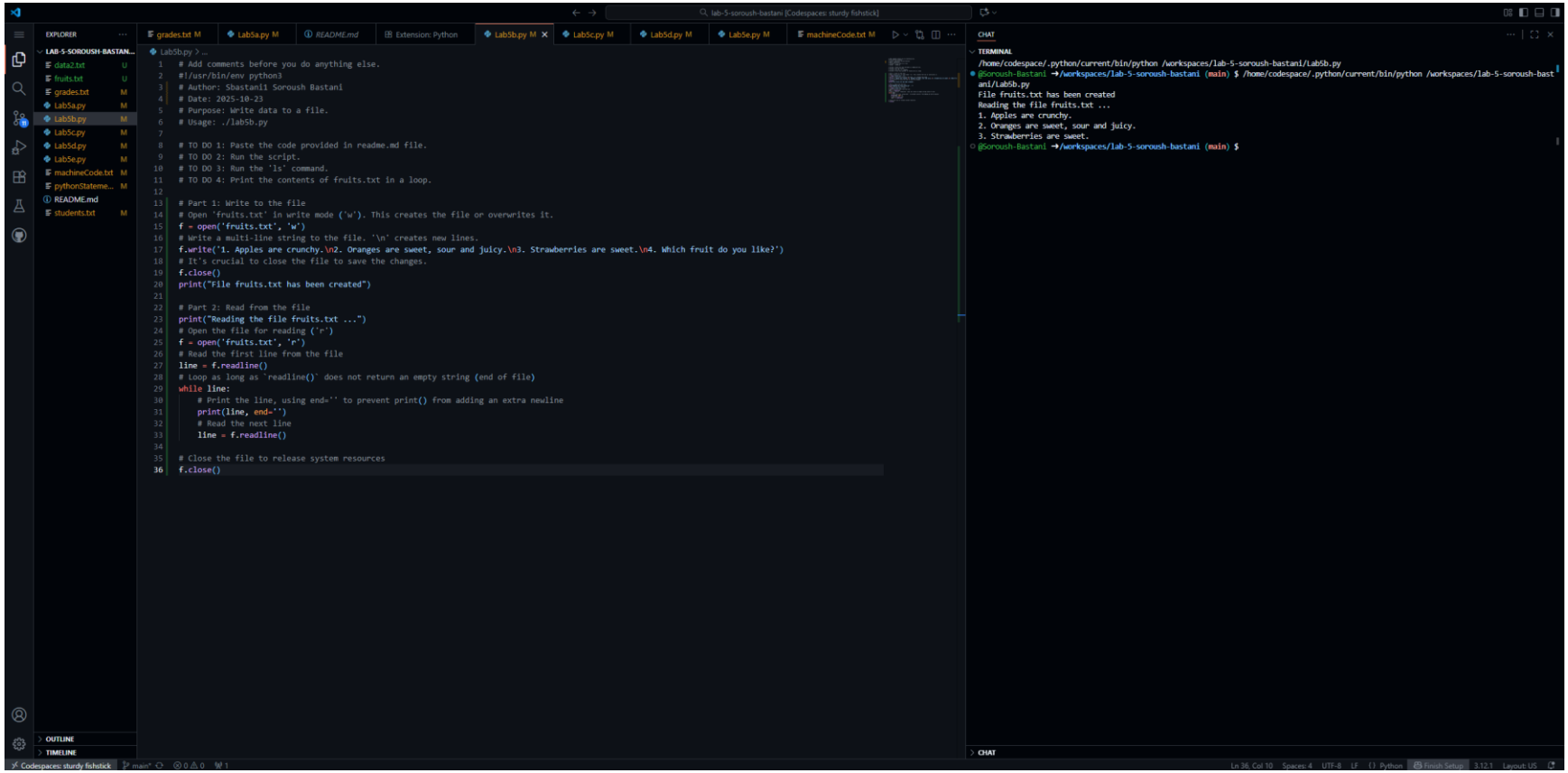
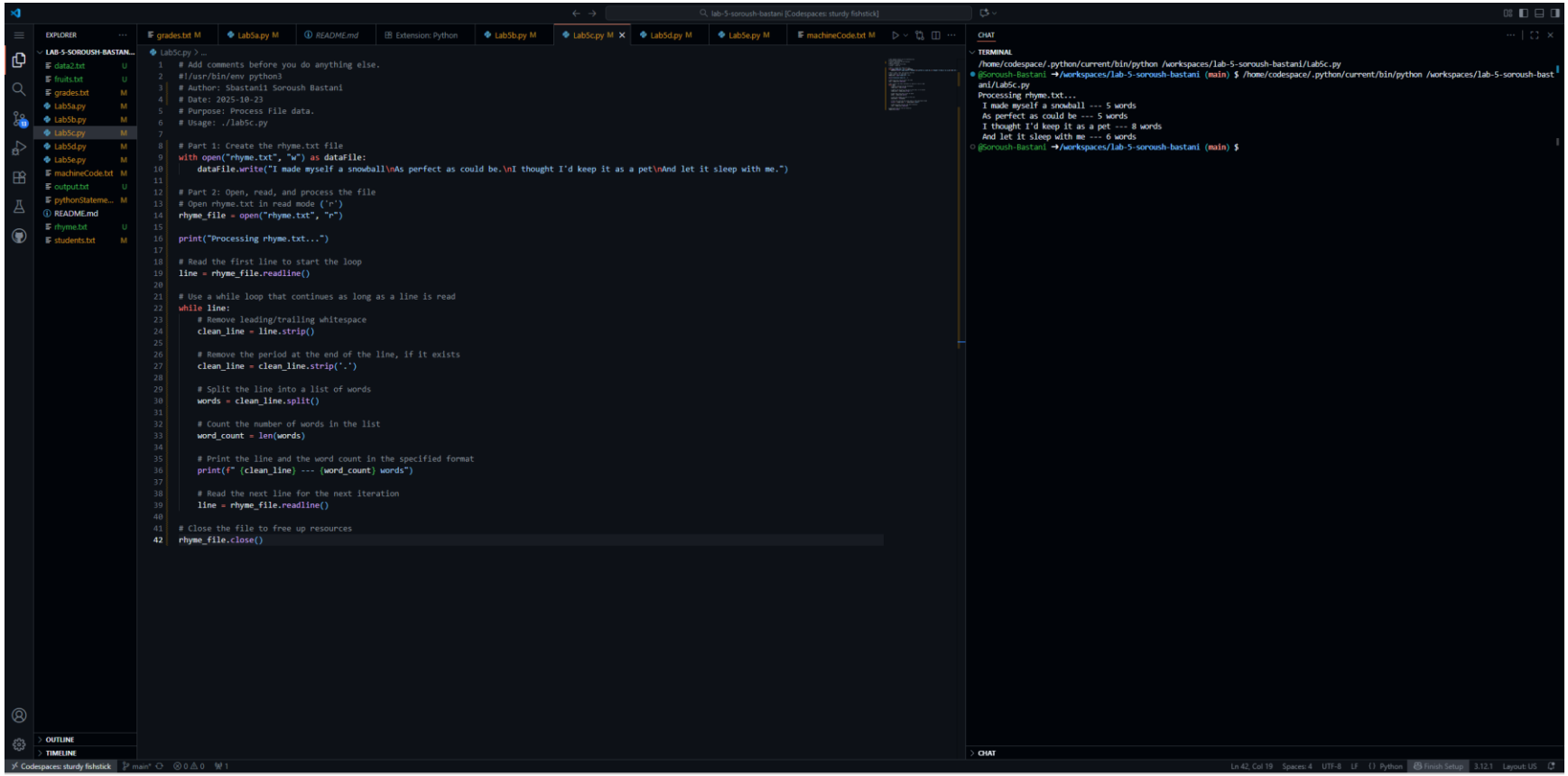
Lab5a


lab-5-soroush-bastani [Codespaces: sturdy fishstick]
EXPLORER
LAB-5-SOROUSH-BASTAN...
data2.txt U
grades.txt M
Lab5a.py M
Lab5b.py M
Lab5c.py M
Lab5d.py M
Lab5e.py M
machineCode.txt M
pythonStateme... M
README.md
students.txt M
Lab5a.py M
README.md
Extension: Python
Lab5b.py M
Lab5c.py M
Lab5a.py > ...
Add comments before you do anything else.
#!/usr/bin/env python3
# Author: Sbastani1 Soroush Bastani
# Date: 2025-10-23
# Purpose: Read a text file.
# Usage: ./lab5a.py
# TO DO 1: Create data2.txt.
# TO DO 2: Complete the functions below.
def read_file_lineByLine(file_name):
"""
Takes file_name as a string, reads the file in a loop
one line at a time and prints each line.
"""
print("Reading file line by line:")
with open(file_name, 'r') as f:
for line in f:
print(line, end='') # Use end='' to avoid double newlines
print("\n" + "="*30) # Separator for clarity
def read_file_list(file_name):
"""
Takes a file_name as a string, reads file line by line and
adds the line to a list as a list element without newline characters.
Returns the list.
"""
print("Reading file into a list of lines:")
with open(file_name, 'r') as f:
# Use a list comprehension to read lines and strip newline characters
lines_list = [line.strip() for line in f]
return lines_list
def main():
"""
Main function to call the file reading functions.
"""
file_name = 'data2.txt'
read_file_lineByLine(file_name)
list_of_lines = read_file_list(file_name)
print(list_of_lines)
if __name__ == "__main__":
main()
OUTLINE
TIMELINE
CHAT
TERMINAL
/home/codespace/.python/current/bin/python /workspaces/lab-5-soroush-bastani/Lab5a.py
@Soroush-Bastani ➜/workspaces/lab-5-soroush-bastani (main) $ /home/codespa
ce/.python/current/bin/python /workspaces/lab-5-soroush-bastani/Lab5a.py
Reading file line by line:
Hello World, Welcome to File Handling!
Line 2
Line 3
Line 4
Line 5
==============================
Reading file into a list of lines:
['Hello World, Welcome to File Handling!', 'Line 2', 'Line 3', 'Line 4', 'Line 5']
@Soroush-Bastani ➜/workspac
@Soroush-Bastani ➜/workspac
@Soroush-Bastani ➜/workspac
@Soroush-Bastani ➜/workspaces/lab-5-soroush-bastani (main) $
CHAT
Codespaces: sturdy fishstick
main*
Ln 42, Col 27
Spaces: 4
UTF-8
LF
Python
Finish Setup
3.12.1
Layout: US

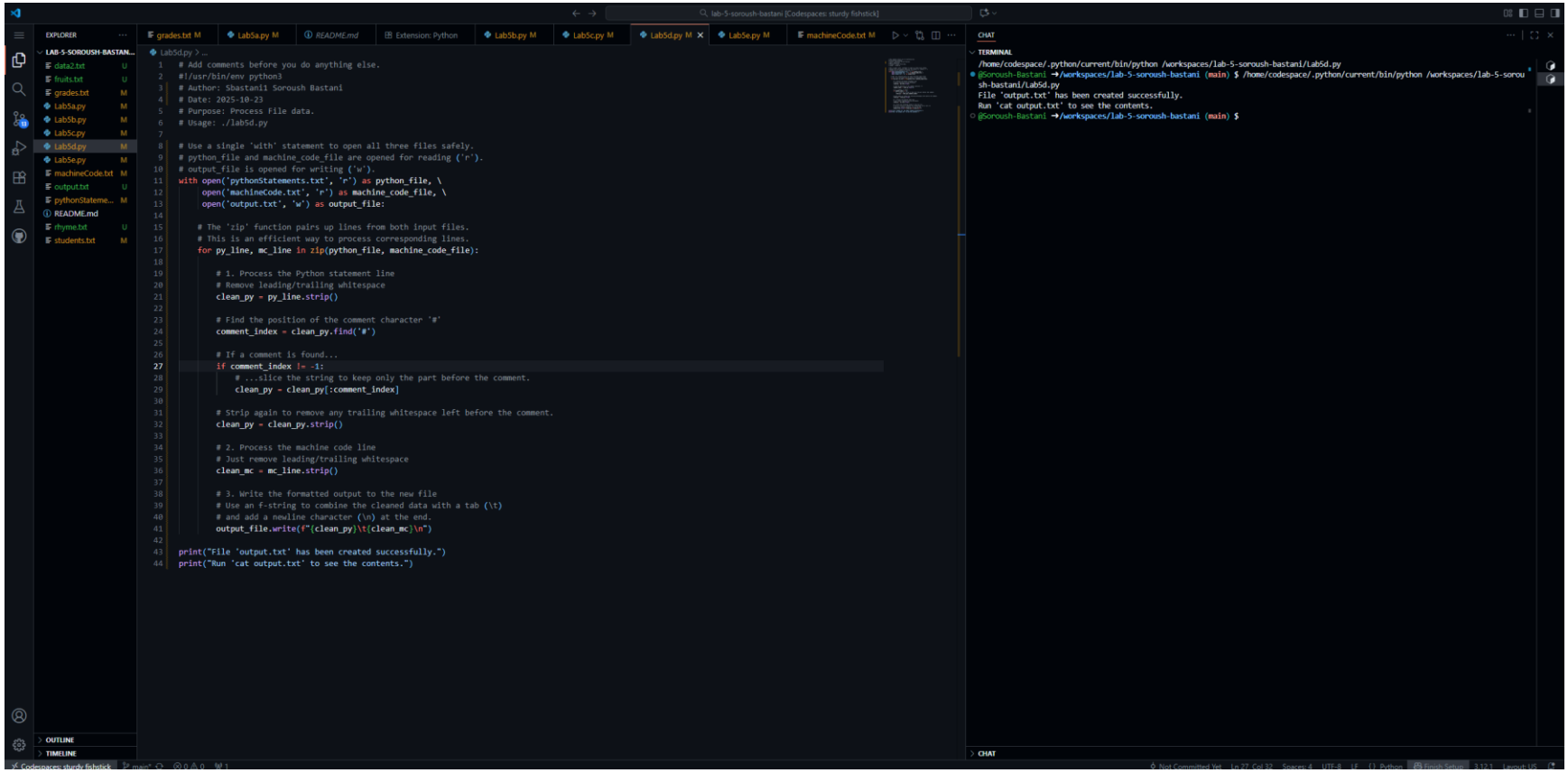
Lab5b


EXPLORER
LAB-5-SOROUSH-BASTAN...
data2.txt
fruits.txt
grades.txt
Lab5a.py
Lab5b.py
Lab5c.py
Lab5d.py
Lab5e.py
machineCode.txt
pythonStateme...
README.md
students.txt
OUTLINE
TIMELINE
grades.txt M
Lab5a.py M
README.md
Extension: Python
Lab5b.py M
Lab5c.py M
Lab5d.py M
Lab5e.py M
machineCode.txt M
Lab5b.py > ...
Add comments before you do anything else.
#!/usr/bin/env python3
# Author: Sbastani1 Soroush Bastani
# Date: 2025-10-23
# Purpose: Write data to a file.
# Usage: ./lab5b.py
# TO DO 1: Paste the code provided in readme.md file.
# TO DO 2: Run the script.
# TO DO 3: Run the 'ls' command.
# TO DO 4: Print the contents of fruits.txt in a loop.
# Part 1: Write to the file
# Open 'fruits.txt' in write mode ('w'). This creates the file or overwrites it.
f = open('fruits.txt', 'w')
# Write a multi-line string to the file. '\n' creates new lines.
f.write('1. Apples are crunchy.\n2. Oranges are sweet, sour and juicy.\n3. Strawberries are sweet.\n4. Which fruit do you like?')
# It's crucial to close the file to save the changes.
f.close()
print("File fruits.txt has been created")
# Part 2: Read from the file
print("Reading the file fruits.txt ...")
# Open the file for reading ('r')
f = open('fruits.txt', 'r')
# Read the first line from the file
line = f.readline()
# Loop as long as `readline()` does not return an empty string (end of file)
while line:
# Print the line, using end='' to prevent print() from adding an extra newline
print(line, end='')
# Read the next line
line = f.readline()
# Close the file to release system resources
f.close()
CHAT
TERMINAL
/home/codespace/.python/current/bin/python /workspaces/lab-5-soroush-bastani/Lab5b.py
@Soroush-Bastani ➜/workspaces/lab-5-soroush-bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-5-soroush-bastani/Lab5b.py
File fruits.txt has been created
Reading the file fruits.txt ...
1. Apples are crunchy.
2. Oranges are sweet, sour and juicy.
3. Strawberries are sweet.
@Soroush-Bastani ➜/workspaces/lab-5-soroush-bastani (main) $
CHAT
Codespaces: sturdy fishstick
main*
Ln 36, Col 10
Spaces: 4
UTF-8
LF
Python
Finish Setup
3.12.1
Layout: US

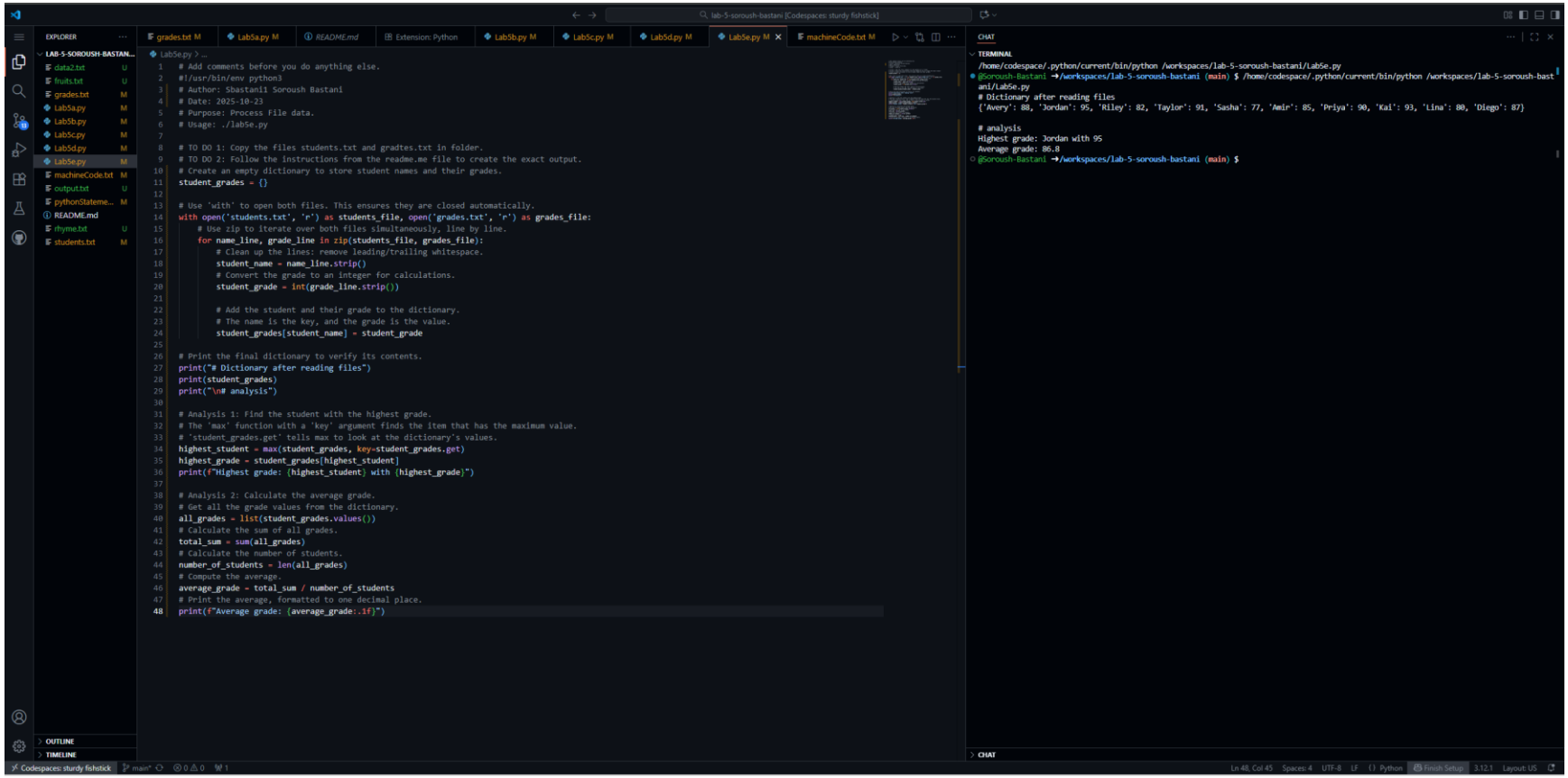
Lab5c


EXPLORER
LAB-5-SOROUSH-BASTAN...
data2.txt
fruits.txt
grades.txt
Lab5a.py
Lab5b.py
Lab5c.py
Lab5d.py
Lab5e.py
machineCode.txt
output.txt
pythonStateme...
README.md
rhyme.txt
students.txt
OUTLINE
TIMELINE
grades.txt M
Lab5a.py M
README.md
Extension: Python
Lab5b.py M
Lab5c.py M
Lab5d.py M
Lab5e.py M
machineCode.txt M
Lab5c.py > ...
Add comments before you do anything else.
#!/usr/bin/env python3
Author: Sbastani1 Soroush Bastani
Date: 2025-10-23
Purpose: Process File data.
Usage: ./lab5c.py
Part 1: Create the rhyme.txt file
with open("rhyme.txt", "w") as dataFile:
dataFile.write("I made myself a snowball\nAs perfect as could be.\nI thought I'd keep it as a pet\nAnd let it sleep with me.")
# Part 2: Open, read, and process the file
# Open rhyme.txt in read mode ('r')
rhyme_file = open("rhyme.txt", "r")
print("Processing rhyme.txt...")
# Read the first line to start the loop
line = rhyme_file.readline()
# Use a while loop that continues as long as a line is read
while line:
# Remove leading/trailing whitespace
clean_line = line.strip()
# Remove the period at the end of the line, if it exists
clean_line = clean_line.strip('.')
# Split the line into a list of words
words = clean_line.split()
# Count the number of words in the list
word_count = len(words)
# Print the line and the word count in the specified format
print(f" {clean_line} --- {word_count} words")
# Read the next line for the next iteration
line = rhyme_file.readline()
# Close the file to free up resources
rhyme_file.close()
CHAT
TERMINAL
/home/codespace/.python/current/bin/python /workspaces/lab-5-soroush-bastani/Lab5c.py
@Soroush-Bastani ➜/workspaces/lab-5-soroush-bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-5-soroush-bastani/Lab5c.py
Processing rhyme.txt...
I made myself a snowball --- 5 words
As perfect as could be --- 5 words
I thought I'd keep it as a pet --- 8 words
And let it sleep with me --- 6 words
@Soroush-Bastani ➜/workspaces/lab-5-soroush-bastani (main) $
CHAT
Codespaces: sturdy fishstick
main*
Ln 42, Col 19
Spaces: 4
UTF-8
LF
Python
Finish Setup
3.12.1
Layout: US

Lab5d


EXPLORER
LAB-5-SOROUSH-BASTAN...
data2.txt
fruits.txt
grades.txt
Lab5a.py
Lab5b.py
Lab5c.py
Lab5d.py
Lab5e.py
machineCode.txt
output.txt
pythonStateme...
README.md
rhyme.txt
students.txt
OUTLINE
TIMELINE
grades.txt M
Lab5a.py M
README.md
Extension: Python
Lab5b.py M
Lab5c.py M
Lab5d.py M
Lab5e.py M
machineCode.txt M
Lab5d.py > ...
Add comments before you do anything else.
#!/usr/bin/env python3
Author: Sbastani1 Soroush Bastani
Date: 2025-10-23
Purpose: Process File data.
Usage: ./lab5d.py
Use a single 'with' statement to open all three files safely.
python_file and machine_code_file are opened for reading ('r').
output_file is opened for writing ('w').
with open('pythonStatements.txt', 'r') as python_file, \
open('machineCode.txt', 'r') as machine_code_file, \
open('output.txt', 'w') as output_file:
# The 'zip' function pairs up lines from both input files.
# This is an efficient way to process corresponding lines.
for py_line, mc_line in zip(python_file, machine_code_file):
# 1. Process the Python statement line
# Remove leading/trailing whitespace
clean_py = py_line.strip()
# Find the position of the comment character '#'
comment_index = clean_py.find('#')
# If a comment is found...
if comment_index != -1:
# ...slice the string to keep only the part before the comment.
clean_py = clean_py[:comment_index]
# Strip again to remove any trailing whitespace left before the comment.
clean_py = clean_py.strip()
# 2. Process the machine code line
# Just remove leading/trailing whitespace
clean_mc = mc_line.strip()
# 3. Write the formatted output to the new file
# Use an f-string to combine the cleaned data with a tab (\t)
# and add a newline character (\n) at the end.
output_file.write(f"{clean_py}\t{clean_mc}\n")
print("File 'output.txt' has been created successfully.")
print("Run 'cat output.txt' to see the contents.")
CHAT
TERMINAL
/home/codespace/.python/current/bin/python /workspaces/lab-5-soroush-bastani/Lab5d.py
@Soroush-Bastani ➜ /workspaces/lab-5-soroush-bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-5-soroush-bastani/Lab5d.py
File 'output.txt' has been created successfully.
Run 'cat output.txt' to see the contents.
@Soroush-Bastani ➜ /workspaces/lab-5-soroush-bastani (main) $
CHAT

Lab5e


lab-5-soroush-bastani [Codespaces: sturdy fishstick]
EXPLORER
LAB-5-SOROUSH-BASTAN...
data2.txt
fruits.txt
grades.txt
Lab5a.py
Lab5b.py
Lab5c.py
Lab5d.py
Lab5e.py
machineCode.txt
output.txt
pythonStateme...
README.md
rhyme.txt
students.txt
OUTLINE
TIMELINE
grades.txt M
Lab5a.py M
README.md
Extension: Python
Lab5b.py M
Lab5c.py M
Lab5d.py M
Lab5e.py M
machineCode.txt M
Lab5e.py > ...
Add comments before you do anything else.
#!/usr/bin/env python3
# Author: Sbastani1 Soroush Bastani
# Date: 2025-10-23
# Purpose: Process File data.
# Usage: ./lab5e.py
# TO DO 1: Copy the files students.txt and gradtes.txt in folder.
# TO DO 2: Follow the instructions from the readme.me file to create the exact output.
# Create an empty dictionary to store student names and their grades.
student_grades = {}
# Use 'with' to open both files. This ensures they are closed automatically.
with open('students.txt', 'r') as students_file, open('grades.txt', 'r') as grades_file:
# Use zip to iterate over both files simultaneously, line by line.
for name_line, grade_line in zip(students_file, grades_file):
# Clean up the lines: remove leading/trailing whitespace.
student_name = name_line.strip()
# Convert the grade to an integer for calculations.
student_grade = int(grade_line.strip())
# Add the student and their grade to the dictionary.
# The name is the key, and the grade is the value.
student_grades[student_name] = student_grade
# Print the final dictionary to verify its contents.
print("# Dictionary after reading files")
print(student_grades)
print("\n# analysis")
# Analysis 1: Find the student with the highest grade.
# The 'max' function with a 'key' argument finds the item that has the maximum value.
# 'student_grades.get' tells max to look at the dictionary's values.
highest_student = max(student_grades, key=student_grades.get)
highest_grade = student_grades[highest_student]
print(f"Highest grade: {highest_student} with {highest_grade}")
# Analysis 2: Calculate the average grade.
# Get all the grade values from the dictionary.
all_grades = list(student_grades.values())
# Calculate the sum of all grades.
total_sum = sum(all_grades)
# Calculate the number of students.
number_of_students = len(all_grades)
# Compute the average.
average_grade = total_sum / number_of_students
# Print the average, formatted to one decimal place.
print(f"Average grade: {average_grade:.1f}")
CHAT
TERMINAL
/home/codespace/.python/current/bin/python /workspaces/lab-5-soroush-bastani/Lab5e.py
@Soroush-Bastani ➜/workspaces/lab-5-soroush-bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-5-soroush-bastani/Lab5e.py
Dictionary after reading files
{'Avery': 88, 'Jordan': 95, 'Riley': 82, 'Taylor': 91, 'Sasha': 77, 'Amir': 85, 'Priya': 90, 'Kai': 93, 'Lina': 80, 'Diego': 87}
analysis
Highest grade: Jordan with 95
Average grade: 86.8
@Soroush-Bastani ➜/workspaces/lab-5-soroush-bastani (main) $
CHAT
Codespaces: sturdy fishstick
main*
Ln 48, Col 45
Spaces: 4
UTF-8
LF
Python
Finish Setup
3.12.1
Layout: US