# ASSIGNMENT 13.3

# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_BATCH:06

# NAME:J.ABHIRAM

**ROLL NO:2403a51342**

**TASK:1**

**Prompt:**

Remove Repetition - Refactor redundant area calculation code.

## Code:



## Output:

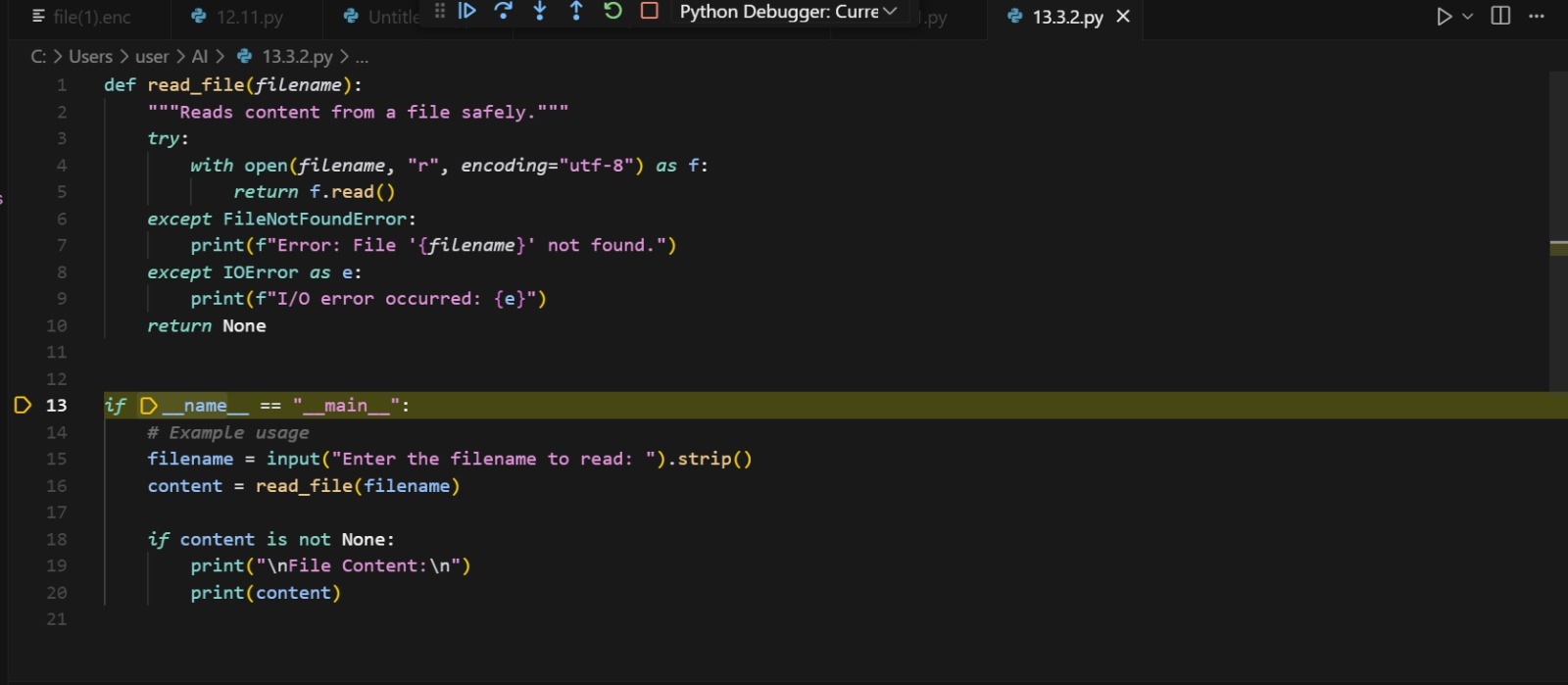
## Observation:The code is modular, avoids repetition, and is easy to extend. Using dictionary dispatch makes the function more scalable. New shapes can be added without modifying existing logic.

# Task 2

## Prompt:

Error Handling in Legacy Code - Improve file reading function.

## Code:



## Output:

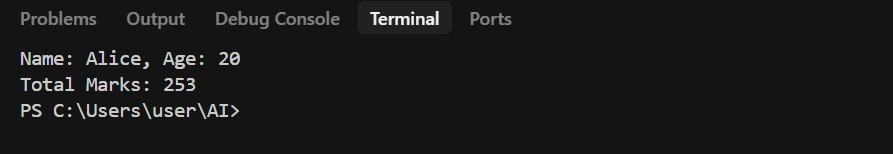
## Observation:The refactored code is safer and prevents crashes if the file is missing. Using 'with open()' ensures automatic file closure. Error handling provides user-friendly feedback.

## Task 3

## Prompt:

Complex Refactoring - Improve Student class readability.

## Code:

**Output:**

## Observation:

The refactored class improves clarity and maintainability. Storing marks in a list makes the design more flexible. The use of docstrings improves code documentation.

## TASK 4

## Prompt:

Inefficient Loop Refactoring - Replace loop with list comprehension.

## Code and Output:

## 

**Observation:**The list comprehension makes code concise and pythonic. It reduces the number of lines of code while improving readability. It is also faster for large datasets compared to append in a loop.