INTRODUCTION

In the modern educational system, peer reviews are an effective method to improve learning outcomes by allowing students to evaluate each other's work. This system not only helps in better understanding the subject but also encourages critical thinking, fairness, and constructive feedback. This Python-based peer review system allows students to submit assignments and review others' submissions.

MISSION

- Facilitate peer evaluation and learning.
- Enable students to receive feedback on their work from their peers.
- Allow students to engage in collaborative learning.
- Provide a fair and transparent evaluation process by aggregating peer feedback.

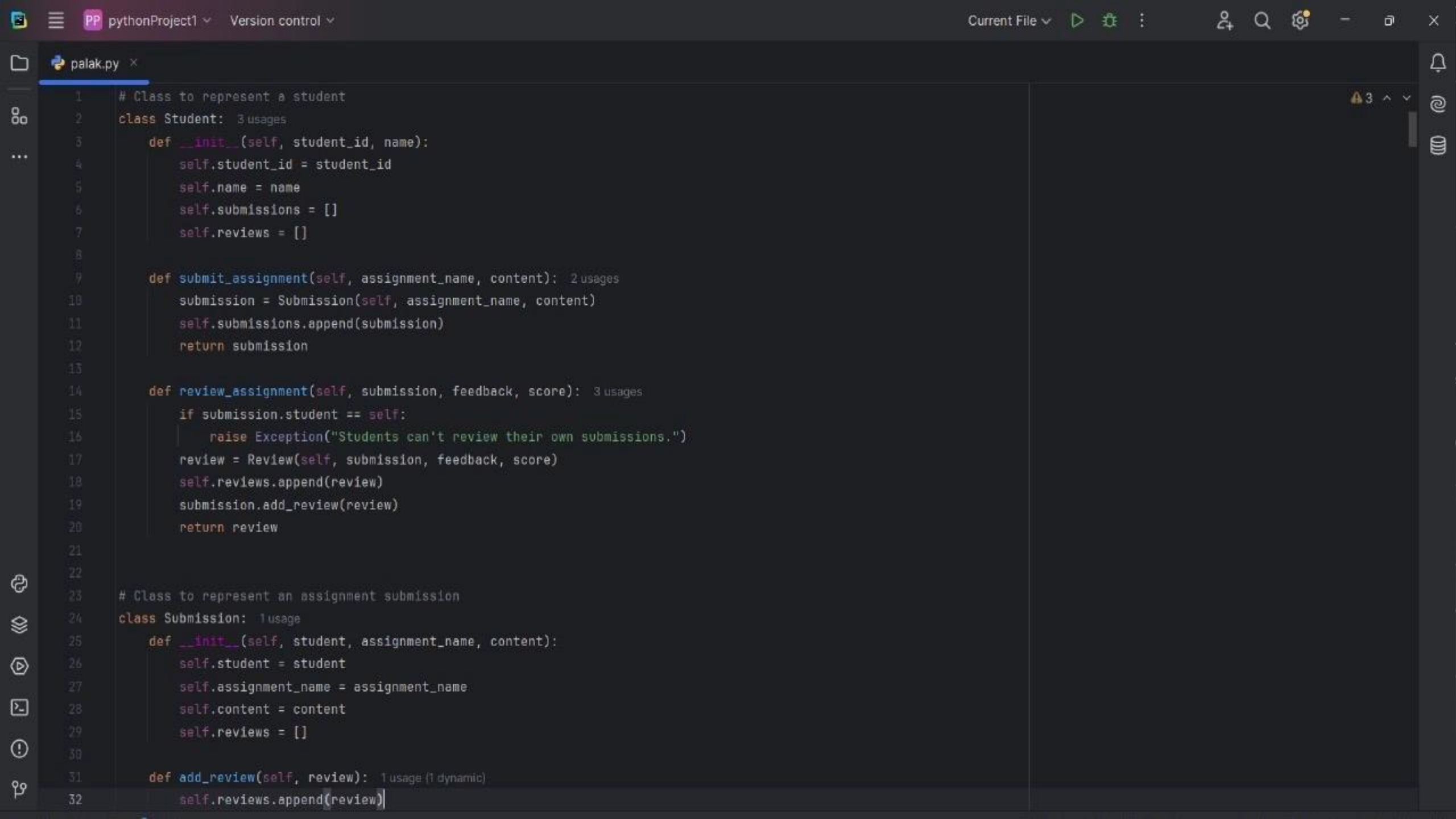
ABSTRACT

This peer review system is designed to allow students to submit assignments and receive reviews from their peers. Each review contains feedback and a score that contributes to an overall grade for the assignment. The system ensures that students cannot review their own submissions and that multiple reviews contribute to a fair assessment. The system supports the following operations:

- Students submit assignments.
- Peers review the submissions.
- The system calculates the average score for each submission.
- The system ensures fairness and prevents self-review.

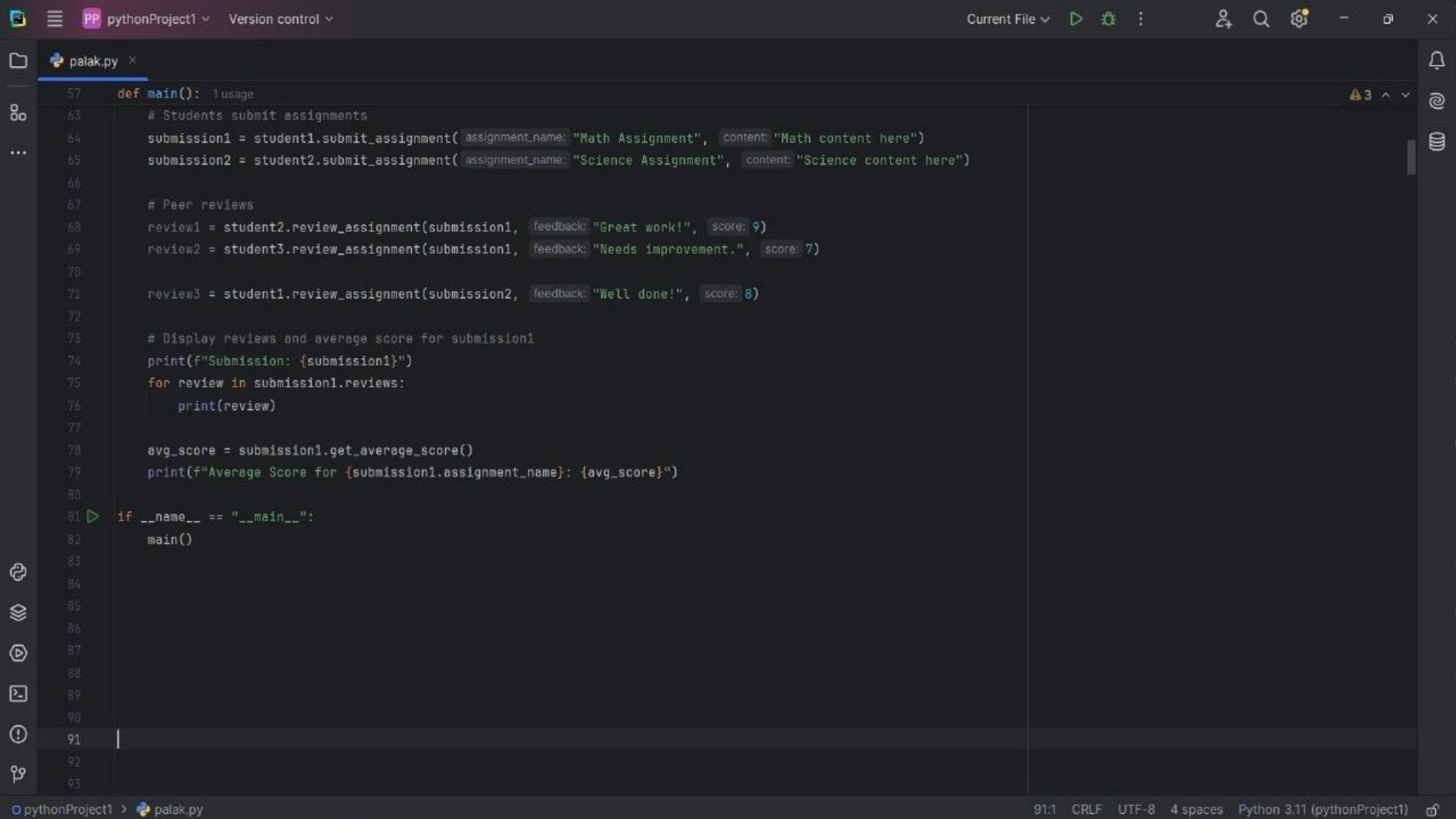
ALGORITHM

1. **Student Submission**: Each student submits an assignment to the system.
2. **Peer Review**: Students review the submissions of their peers, providing feedback and a score.
3. **Validation**: The system ensures that no student can review their own submission.
4. **Score Calculation**: The system computes the average score based on the peer reviews.
5. **Feedback Display**: The system displays the feedback and score to the student.


pythonProject1
Version control
Current File
palak.py
Class to represent a student
class Student: 3 usages
def __init__(self, student_id, name):
self.student_id = student_id
self.name = name
self.submissions = []
self.reviews = []
def submit_assignment(self, assignment_name, content): 2 usages
submission = Submission(self, assignment_name, content)
self.submissions.append(submission)
return submission
def review_assignment(self, submission, feedback, score): 3 usages
if submission.student == self:
raise Exception("Students can't review their own submissions.")
review = Review(self, submission, feedback, score)
self.reviews.append(review)
submission.add_review(review)
return review
Class to represent an assignment submission
class Submission: 1 usage
def __init__(self, student, assignment_name, content):
self.student = student
self.assignment_name = assignment_name
self.content = content
self.reviews = []
def add_review(self, review): 1 usage (1 dynamic)
self.reviews.append(review)
3

palak.py

```
class Submission:  1 usage
        self.reviews.append(review)

    def get_average_score(self):  1 usage
        if not self.reviews:
            return None
        total_score = sum([review.score for review in self.reviews])
        return total_score / len(self.reviews)

    def __str__(self):
        return f"Assignment: {self.assignment_name}, Submitted by: {self.student.name}"


# Class to represent a peer review
class Review:  1 usage
    def __init__(self, reviewer, submission, feedback, score):
        self.reviewer = reviewer
        self.submission = submission
        self.feedback = feedback
        self.score = score

    def __str__(self):
        return f"Reviewer: {self.reviewer.name}, Score: {self.score}, Feedback: {self.feedback}"


# Example Usage
def main():  1 usage
    # Create students
    student1 = Student( student_id: 1,  name: "Alice")
    student2 = Student( student_id: 2,  name: "Bob")
    student3 = Student( student_id: 3,  name: "Charlie")
```


pythonProject1
Version control
Current File
palak.py
def main(): 1 usage
# Students submit assignments
submission1 = student1.submit_assignment(assignment_name: "Math Assignment", content: "Math content here")
submission2 = student2.submit_assignment(assignment_name: "Science Assignment", content: "Science content here")
# Peer reviews
review1 = student2.review_assignment(submission1, feedback: "Great work!", score: 9)
review2 = student3.review_assignment(submission1, feedback: "Needs improvement.", score: 7)
review3 = student1.review_assignment(submission2, feedback: "Well done!", score: 8)
# Display reviews and average score for submission1
print(f"Submission: {submission1}")
for review in submission1.reviews:
print(review)
avg_score = submission1.get_average_score()
print(f"Average Score for {submission1.assignment_name}: {avg_score}")
if __name__ == "__main__":
main()
pythonProject1 > palak.py
91:1 CRLF UTF-8 4 spaces Python 3.11 (pythonProject1)

pythonProject1 Version control Current File

palak.py

```
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
```

Run palak

```
C:\Users\Admin\PycharmProjects\pythonProject1\.venv\Scripts\python.exe C:\Users\Admin\PycharmProjects\pythonProject1\palak.py
Submission: Assignment: Math Assignment, Submitted by: Alice
Reviewer: Bob, Score: 9, Feedback: Great work!
Reviewer: Charlie, Score: 7, Feedback: Needs improvement.
Average Score for Math Assignment: 8.0

Process finished with exit code 0
```

CONCLUSION

This student peer review system provides a straightforward approach to peer-based assignment evaluation.
It ensures fairness and constructive feedback by involving multiple students in the review process.
The Python implementation is modular and can be easily extended with additional features, such as grading
rubrics or more complex feedback mechanisms.

SYSTEM REQUIREMENTS

-HARDWARE REQUIREMENTS

Minimum Hardware:

- **Processor**: Intel Core i3 or equivalent (1.6 GHz or faster)
- **RAM**: 4 GB (8 GB recommended for smoother operation)
- **Storage**: At least 500 MB of free space for Python, dependencies, and data storage
- **Display**: 1366x768 screen resolution
- **Internet Connection**: Required for installing dependencies and libraries

-SOFTWARE REQUIREMENTS

Operating System:

- **Windows 10/11**
- **macOS 10.15 (Catalina)** or newer
- **Linux** (Ubuntu, Fedora, etc.)

Python Version:

- **Python 3.7 or later** (preferably Python 3.10 or newer for improved performance and features)

REFRENCES

- GOOGLE CHROME
- CHAT GPT

GROUP PROJECT BY-

- PALAK S SACHAR - 3BR23CS114
- SANIA SHAIKH - 3BR23CS136
- SANIYA HUNDEKAR - 3BR23CS137
- S VAISHNAVI - 3BR23CS134
- LAYASHREE L - 3BR23CS084