Problem statement on containment

Write a program to

1. **create a package “SY” which has class SYMARKS (Computer Total, MathsTotal, ElectronicsTotal).**

```
# SY/symarks.py
class SYMARKS:
    def __init__(self, computer_total, maths_total, electronics_total):
        self.computer_total = computer_total
        self.maths_total = maths_total
        self.electronics_total = electronics_total

    def get_computer_total(self):
        return self.computer_total
```

2. **Create another package “TY” which has a class TYMarks (Theory, Practical).**

```
# TY/tymarks.py
class TYMarks:
    def __init__(self, theory, practical):
        self.theory = theory
        self.practical = practical

    def get_total(self):
        return self.theory + self.practical
```

3. **Create object of student class (Outside SY & TY package) having roll number, name, SYMakrs and TYMarks. Add the marksof SY and TY Computer subjects and calculate grade ("A" for >=70, "B" for >=60, "C" for >=50, “Pass Class” for >=40 else “Fail”) and display the result of the student in proper format.**

```
# student_main.py
from symarks import SYMARKS
from tymarks import TYMarks

class Student:
    def __init__(self, roll_number, name, sy_marks: SYMARKS, ty_marks: TYMarks):
        self.roll_number = roll_number
```

```
        self.name = name
        self.sy_marks = sy_marks
        self.ty_marks = ty_marks

    def calculate_grade(self):
        # Add Computer marks from SY and TY
        total_computer_marks = self.sy_marks.get_computer_total() +
self.ty_marks.get_total()

        if total_computer_marks >= 70:
            return "A"
        elif total_computer_marks >= 60:
            return "B"
        elif total_computer_marks >= 50:
            return "C"
        elif total_computer_marks >= 40:
            return "Pass Class"
        else:
            return "Fail"

    def display_result(self):
        print(f"Student Roll No.: {self.roll_number}")
        print(f"Student Name: {self.name}")
        print(f"SY Computer Total: {self.sy_marks.get_computer_total()}")
        print(f"TY Computer Total (Theory + Practical): {self.ty_marks.get_total()}")
        print(f"Total Computer Marks: {self.sy_marks.get_computer_total() +
self.ty_marks.get_total()}")
        print(f"Grade: {self.calculate_grade()}")

# Example usage
if __name__ == "__main__":
    # Create SY marks object
    sy_marks = SYMARKS(computer_total=40, maths_total=50, electronics_total=45)

    # Create TY marks object (theory + practical)
    ty_marks = TYMarks(theory=30, practical=25)

    # Create student object
    student = Student(roll_number=101, name="Alice", sy_marks=sy_marks,
ty_marks=ty_marks)

    # Display student result
    student.display_result()
```