**Task 1: Student Grades Management**

**Description:**  
Create a class Student to manage student details and grades using **encapsulation**.

**Requirements:**

1. Private attributes:
   * \_\_name (string)
   * \_\_grades (list of integers)
2. Public methods:
   * add\_grade(grade) → adds a grade to the list
   * get\_average() → returns the average of grades
   * get\_name() → returns the student’s name
3. Rules:
   * Direct access to \_\_grades or \_\_name should **not** be allowed.

**Example Usage:**

student = Student("Alice")

student.add\_grade(85)

student.add\_grade(90)

print(student.get\_average()) # Output: 87.5

print(student.get\_name()) # Output: Alice

**Task 2: Employee Salary Management**

**Description:**  
Create a class Employee to manage employee salary using **encapsulation**.

**Requirements:**

1. Private attributes:
   * \_\_name (string)
   * \_\_salary (float)
2. Public methods:
   * set\_salary(amount) → sets salary only if it is positive
   * get\_salary() → returns salary
   * get\_name() → returns employee name
3. Rules:
   * Trying to set a negative salary should print: "Invalid salary!"
   * Direct access to \_\_salary and \_\_name should **not** be allowed.

**Example Usage:**

emp = Employee("Bob")

emp.set\_salary(5000)

print(emp.get\_salary()) # Output: 5000

emp.set\_salary(-100) # Output: Invalid salary!

print(emp.get\_name()) # Output: Bob