**2023 Spring OOP II**

**HW\_4**

**Q.1**

Write a decorator to compare the time taken to compute Fibonacci(35) by the following 3 functions:

(1) recursive Fibonacci with no memoization

(2) recursive Fibonacci with memoization

(3) Fibonacci via a generator

**Q.2**

You are working on a project to create a basic student management system for a school. You are tasked with creating a SQLite3 database using Python to store and manage student records. The database should have the following functionalities:

(1) Create a table named "students" with the following columns: student\_id (integer), first\_name (text),   
 last\_name (text), age (integer), and grade (text).

(2) Write a Python function that adds a new student record to the "students" table with the given inputs:   
 first name, last name, age, and grade.

(3) Write a Python function that retrieves all the student records from the "students" table and displays  
 them in a formatted manner.

(4) Write a Python function that allows the user to search for a student record by their student ID   
 and displays the record if found.

(5) Write a Python function that allows the user to update the age of a student by their student ID.

(6) Write a Python function that allows the user to delete a student record by their student ID.

Your task is to implement the above functionalities using SQLite3 and Python. Include appropriate error handling and make sure to close the database connection after each operation.

Start with inserting the following 5 students into the "students" table:

**First Name Last Name Age Grade**

**===================================================**

**John Doe 18 Grade 12**

**Jonathan Cole 16 Grade 10**

**Jane Smith 17 Grade 11**

**Elisabeth Shue 17 Grade 11**

**David Lee 16 Grade 10**

Note: You can assume that the student\_id column is unique and auto-incrementing.

**Q.3**

Write a string reverser using a generator in Python. For the input string

**Roger Federer: GOAT**

the output is

**TAOG :reredeF regoR**

**Q.4 Stack Implementations**

(1) Using inheritance (is-a relationship), implement your own stack class by extending the built-in list.

(2) Using composition (has-a relationship), implement your own stack class by creating a list as   
 an attribute (or a data member/field).

**Q.5**

You are given a list of integers. Write a Python program that generates a new list of integers using a generator function. The generator should only include even numbers from the original list, and each even number should be squared.

Example input: **[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]**

Example output: **[4, 16, 36, 64, 100]**

**Q.6**

Graphical user interface, text, application, email

Description automatically generated