**Programming Assignment Unit 7.**

In this assignment, I will combine my knowledge of Python functions, loops, and data structures (dictionaries) to write code of a dictionary that contains lists of students and their courses. I will take a look at different concepts that can be used to manipulate the data(for example with key values).

The Python Code:

# Original dictionary (I will name it stud\_dict) with students as keys and their courses as values

stud\_dict = {

'Stud1': ['CS1101', 'CS2402', 'CS2001'],

'Stud2': ['CS2402','CS2001','CS1102']

}

# Function to invert the dictionary

def invert\_dict(stud\_dict):

inverted\_dict = {}

for student, courses in stud\_dict.items():

for course in courses:

if course in inverted\_dict:

inverted\_dict[course].append(student)

else:

inverted\_dict[course] = [student]

return inverted\_dict

# Inverted dictionary with courses as keys and students as values

inverted\_dict = invert\_dict(stud\_dict)

# Print the original dictionary

print("Original Dictionary: ", stud\_dict)

# Print the inverted dictionary

print("Inverted Dictionary: ", inverted\_dict)

output:

Original Dictionary:

{'Stud1': ['CS1101', 'CS2402', 'CS2001'],

'Stud2': ['CS2402', 'CS2001', 'CS1102']}

Inverted Dictionary:

{'CS1101': ['Stud1'],

'CS2402': ['Stud1', 'Stud2'],

'CS2001': ['Stud1', 'Stud2'],

'CS1102': ['Stud2']}

The Explanation:

In the above python code, the funtion invert\_dict takes a dictionary stud\_dict as input where each key is a student and the value is a list of courses that the student is enrolled in. It then iterates over each student-course pair in the input dictionary.

For each course in the list of courses, it checks if the course is already a key in the inverted\_dict. If it is, it appends the student to the list of students for that course. If it’s not, it creates a new key-value pair in the inverted\_dict with the course as the key and a new list containing the student as the value.

In the output of this function, there will be a new dictionary where each key is a course and the value is a list of students enrolled in that course. I print the inverted dictionary along with the original dictionary.

This function effectively shows the relationship between students and courses. This makes it easy to find all the students enrolled in a particular course. Also given the fact that dictionary keys in Python are unique, each course can only appear once as a key in the inverted dictionary.

The student list lets us connect many students to each course. This process takes O(n) time (time complexity), where n is the total number of student-course pairs because it goes through each pair once. The space needed is also O(n) because it stores each pair in a new dictionary. This method is very efficient even with a lot of students and courses.

#### **References:**

* + Downey, A. (2015). [*Think Python: How to think like a computer scientist*](https://greenteapress.com/thinkpython2/thinkpython2.pdf) (2nd ed.). Green Tea Press.
  + Coreh Schafer. (17 May 2017). Python Tutorial for Beginners 5: Dictionaries- working with key-value pairs [Video file]. Retrieved from <https://youtu.be/daefaLgNkw0?si=6dIo_1P5dsN_0YKP>
  + CS50x .(31 Dec 2023). Lecture 5 -Data Structures [video file]. Retrieved from <https://youtu.be/0euvEdPwQnQ?si=kX_2PSL1hLUVyyrU>