Programming Assignment Uni 7

Irfan Manzoor

University of the People

CS 1101-01: Programming Fundamentals – AY2025-T1

Alok Parihar

25 October 2024

# Assignment 7

This Python script provides functionality to invert a dictionary representing students and their enrolled courses into a dictionary where courses are keys and students are values, with each course having at most three different students. The `invert\_dictionary` function achieves this inversion by iterating through the original dictionary and populating the inverted dictionary accordingly. Additionally, the script includes a `print\_courses\_and\_students` function to display the courses and their enrolled students. Comments throughout the code provide clarity on the purpose and functionality of each component. The script demonstrates the usage of these functions with an example original dictionary, printing both the original and inverted dictionaries along with their respective courses and students enrolled.

**Code**

# code.png

# Explanation

**Function Definition: `invert\_dictionary(original\_dict)`**

def invert\_dictionary(original\_dict):

This line defines a function named `invert\_dictionary` that takes one argument, `original\_dict`, which is the dictionary to be inverted.

inverted\_dict = {}

Here, an empty dictionary `inverted\_dict` is initialized. This dictionary will hold the inverted data structure where courses are keys and lists of students are values.

for students, courses in original\_dict.items():

This loop iterates over each key-value pair in the `original\_dict`. Each key represents a student, and the corresponding value is a list of courses that the student is enrolled in.

for course in courses:

Within the loop, it iterates over each course in the list of courses for the current student.

if course not in inverted\_dict:

inverted\_dict[course] = []

For each course, it checks if it already exists as a key in the `inverted\_dict`. If not, it initializes it with an empty list. This ensures that each course has an entry in the inverted dictionary.

inverted\_dict[course].append(students)

It appends the current student to the list of students associated with the current course in the `inverted\_dict`.

if len(inverted\_dict[course]) == 3:

break

It checks if the number of students enrolled in the current course has reached three. If so, it breaks out of the loop to ensure that each course in the inverted dictionary has at most three different students.

return inverted\_dict

Finally, it returns the inverted dictionary containing courses as keys and lists of students as values.

**Function Definition: `print\_courses\_and\_students(courses)`**

def print\_courses\_and\_students(courses):

This line defines a function named `print\_courses\_and\_students` that takes one argument, `courses`, which is the dictionary containing courses as keys and lists of students as values.

for course, students in courses.items(): # Iterate over the key-value pairs

Within the function, it iterates over the key-value pairs in the `courses` dictionary, where each key is a course, and the corresponding value is a list of students enrolled in that course.

print(f"Course: {course}") # Print the course

print("Students Enrolled:")

for student in students: # Print the list of students

print(f"- {student}") # Print each student

print()

For each course, it prints the course name and then iterates over the list of students enrolled in that course, printing each student's name preceded by a dash ("-"). After printing all students enrolled in a particular course, it prints a blank line to separate each course's information.

**The original dictionary**

original\_dict = {

"Arfan": ["Computer Science", "Mathematics", "Statistics"],

"Adnan": ["Computer Science", "Data Structures", "UML"],

"Aadil": ["Mathematics", "Statistics", "Data Structures"],

}

In this block we initialize the original\_dict dictionary with some student and courses data. Each key in the dictionary represents a student's name, and the corresponding value is a list of courses that the student is enrolled in.

print("Original Dictionary:")

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

print(f"Student: {student}")

print("Courses Enrolled:")

for course in courses:

print(f"- {course}")

print()

Next we print the contents of the original\_dict. It iterates over each key-value pair in the dictionary using items(). For each student, it prints the student's name and then iterates over the list of courses they are enrolled in. For each course, it prints the course name preceded by a dash ("-"). After printing all courses for a student, it prints a blank line to separate each student's information.

inverted\_dict = invert\_dictionary(original\_dict)

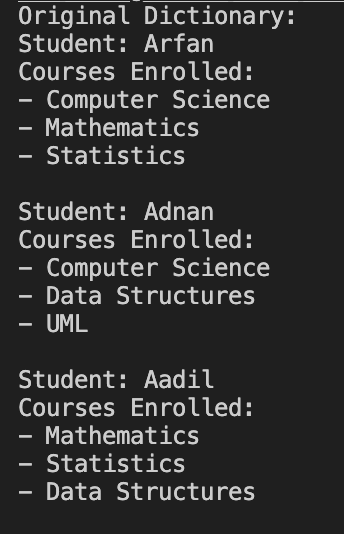
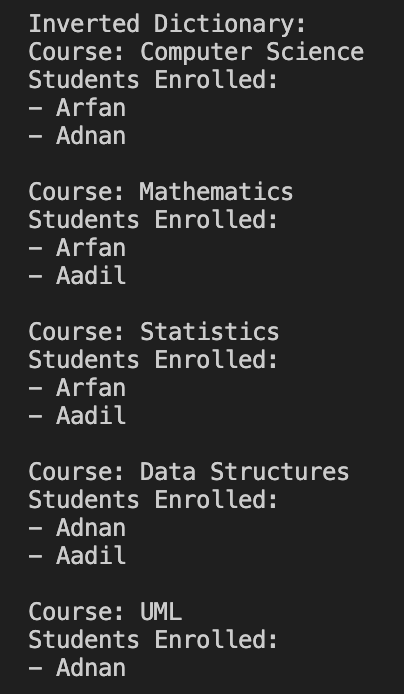
This calls the invert\_dictionary function, passing the original\_dict as an argument. It stores the result, which is the inverted dictionary, in the variable inverted\_dict.

print("\nInverted Dictionary:")

print\_courses\_and\_students(inverted\_dict)

Finally, it prints the contents of the inverted\_dict. It first prints a header indicating that it's printing the inverted dictionary. Then, it calls the print\_courses\_and\_students function, passing the inverted\_dict as an argument. This function prints each course and the students enrolled in that course. Each course's information is printed with the course name followed by the list of enrolled students.

# Output



**References**

Python Software Foundation. (n.d.). *Python 2.12.2 documentation*. Python Official Documentation.

https://docs.python.org/

Python Software Foundation. (n.d.). *Dictionaries: Data structures*. Python Official Documentation.

https://docs.python.org/3/tutorial/datastructures.html#dictionaries

Python Software Foundation. (n.d.). *Control flow tools: if-for tools*. Python Official Documentation.

https://docs.python.org/3/tutorial/controlflow.html

Matthes, E. (2022). *Python crash course* (2nd ed.). No Starch Press.

Sweigart, A. (2019). *Automate the boring stuff with Python* (2nd ed.). No Starch Press.

Ramalho, L. (2015). *Fluent Python*. O’Reilly Media.