Creating Python Programs

# Pseudocode

BEGIN  
• Create a dictionary for course room numbers with keys as course numbers and values as room numbers.  
• Create a dictionary for instructors with keys as course numbers and values as instructor names.  
• Create a dictionary for meeting times with keys as course numbers and values as meeting times.  
• Prompt the user to enter a course number.  
• Retrieve and display the course's room number, instructor, and meeting time based on user input.  
• Handle cases where the course number is not found by displaying an appropriate error message.  
END

# Source Code

def get\_course\_info(course\_number, course\_dicts):

"""

Retrieves course information based on the provided course number from the dictionaries.

Args:

course\_number (str): The course number to look up.

course\_dicts (dict): A dictionary containing dictionaries for room, instructor, and time.

Returns:

tuple: Contains room number, instructor, and meeting time if course is found, otherwise returns an error message.

"""

room\_dict, instructor\_dict, meeting\_time\_dict = course\_dicts

try:

room = room\_dict[course\_number]

instructor = instructor\_dict[course\_number]

meeting\_time = meeting\_time\_dict[course\_number]

return room, instructor, meeting\_time

except KeyError:

return None, f"Error: Course \'{course\_number}\' not found."

def display\_course\_info(course\_number, course\_info):

"""

Displays the course information or an error message.

Args:

course\_number (str): The course number.

course\_info (tuple): Contains course details or an error message.

"""

if course\_info[0]:

room, instructor, meeting\_time = course\_info

print(f"Course: {course\_number}")

print(f"Room Number: {room}")

print(f"Instructor: {instructor}")

print(f"Meeting Time: {meeting\_time}")

else:

print(course\_info[1])

def main():

"""

Main function to run the program logic.

"""

course\_room = {

'CSC101': 3004,

'CSC102': 4501,

'CSC103': 6755,

'NET110': 1244,

'COM241': 1411

}

course\_instructor = {

'CSC101': 'Haynes',

'CSC102': 'Alvarado',

'CSC103': 'Rich',

'NET110': 'Burke',

'COM241': 'Lee'

}

course\_meeting\_time = {

'CSC101': '8:00 a.m.',

'CSC102': '9:00 a.m.',

'CSC103': '10:00 a.m.',

'NET110': '11:00 a.m.',

'COM241': '1:00 p.m.'

}

course\_dicts = (course\_room, course\_instructor, course\_meeting\_time)

while True:

course\_number = input("Enter the course number (e.g., CSC101) or type 'exit' to quit: ")

if course\_number.lower() == 'exit':

print("Exiting the program. Thank you for using it!")

break

course\_number = course\_number.upper()

course\_info = get\_course\_info(course\_number, course\_dicts)

display\_course\_info(course\_number, course\_info)

print("|-------------------------------|")

if \_\_name\_\_ == '\_\_main\_\_':

main()

# Screenshots of Execution

S A screen shot of a computer

Description automatically generated

# Results

The program was tested extensively to ensure its accuracy and reliability. The dictionaries were successfully accessed to retrieve course information, and user inputs were correctly handled, including the display of error messages when necessary. The program effectively demonstrates how to manage and retrieve structured data using Python dictionaries and user input.

# Git Repository

The repository containing the complete source code for this assignment is available [here](https://github.com/Mr-Abe/critical_thinking_assignments/tree/master/Module_7_Assignment).