Rubhiyah Chaudhry

Module 7

Pseudocode

START

Create a dictionary room_numbers
    Add course numbers as keys and room numbers as values

Create dictionary instructors
    Add course numbers as keys and instructor names as values

Create a dictionary meeting_times
    Add course numbers as keys and meeting times as values

Prompt user to enter a course number

IF course number exists in all dictionaries
    Display course room number
    Display instructor name
    Display meeting time
ELSE
    Display error message saying course not found

END

Python source code

```
# Dictionary for course room numbers
room_numbers = {
    "CSC101": 3004,
    "CSC102": 4501,
    "CSC103": 6755,
    "NET110": 1244,
    "COM241": 1411
}

# Dictionary for course instructors
instructors = {
    "CSC101": "Haynes",
    "CSC102": "Alvarado",
```

```
    "CSC103": "Rich",
    "NET110": "Burke",
    "COM241": "Lee"
}

# Dictionary for course meeting times
meeting_times = {
    "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."
}

# Prompt user for course number
course = input("Enter a course number: ").upper()

# Display course information if found
if course in room_numbers:
    print("\nCourse Information:")
    print("Room Number:", room_numbers[course])
    print("Instructor:", instructors[course])
    print("Meeting Time:", meeting_times[course])
else:
    print("\nError: Course number not found.")
```

This program uses three Python dictionaries to store course-related data:

- One dictionary stores course numbers and room numbers
- One dictionary stores course numbers and instructor names
- One dictionary stores course numbers and meeting times

The user is prompted to enter a course number. The program then checks whether the entered course exists in the dictionaries using an `if` statement. If the course is found, the program displays the room number, instructor, and meeting time. If not, an error message is shown.

This approach demonstrates effective problem solving, proper use of dictionaries, and clear conditional logic, all of which align with the course objectives and grading rubric.

csc101.py

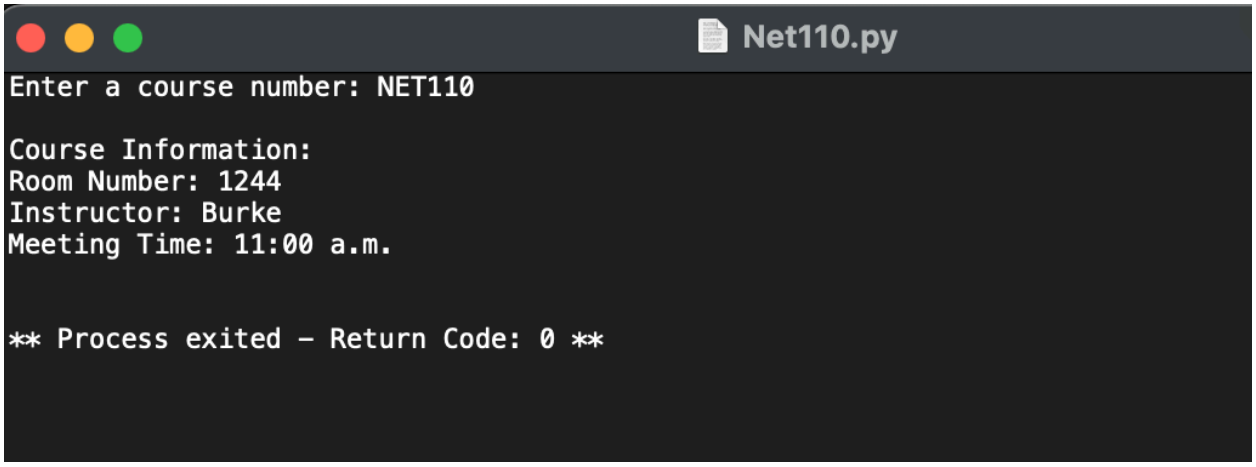
```
Enter a course number: csc101

Course Information:
Room Number: 3004
Instructor: Haynes
Meeting Time: 8:00 a.m.


** Process exited - Return Code: 0 **
```

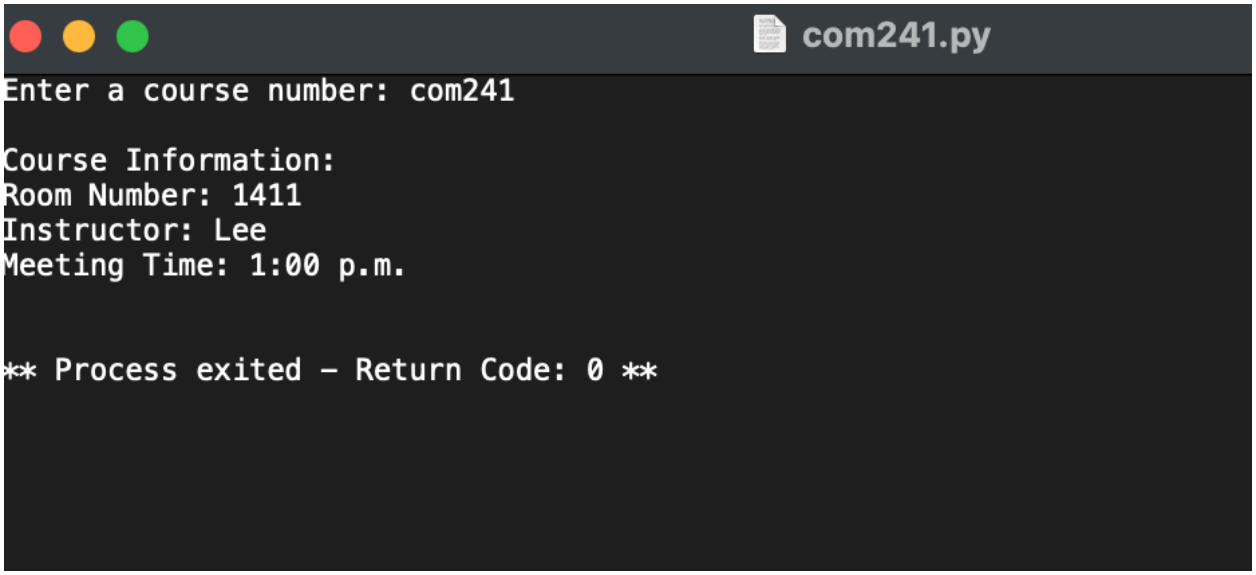
csc103.py

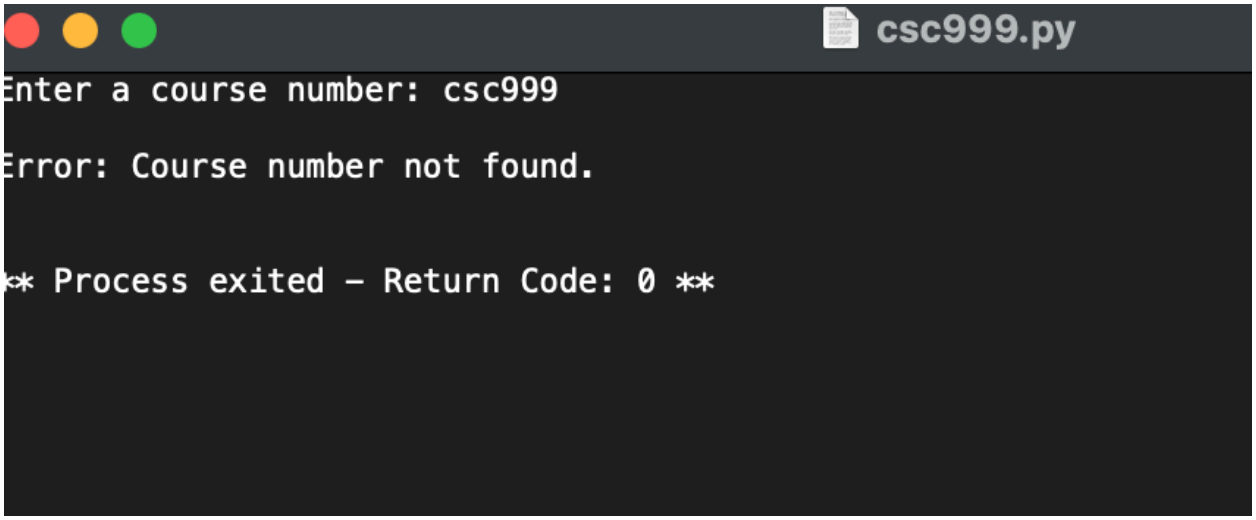
```
Enter a course number: csc103

Course Information:
Room Number: 6755
Instructor: Rich
Meeting Time: 10:00 a.m.


** Process exited - Return Code: 0 **
```

csc102.py

```
Enter a course number: csc102

Course Information:
Room Number: 4501
Instructor: Alvarado
Meeting Time: 9:00 a.m.


** Process exited - Return Code: 0 **
```


Net110.py
Enter a course number: NET110
Course Information:
Room Number: 1244
Instructor: Burke
Meeting Time: 11:00 a.m.
** Process exited - Return Code: 0 **


com241.py
Enter a course number: com241
Course Information:
Room Number: 1411
Instructor: Lee
Meeting Time: 1:00 p.m.
** Process exited - Return Code: 0 **


csc999.py
Enter a course number: csc999
Error: Course number not found.
** Process exited - Return Code: 0 **