**Module7 Critical Thinking Git Repo**

|  |
| --- |
|  |

**source code**

**Problem statement:** University Cources.

**Assumptions:**

1. Course details are static and consider preload them as given in the problem statement.
2. User can continously enter the cource name/s, and they are comma [‘,’ ] sepearted values
3. Program terminates when user enters ‘q’ [quit] option

**Code:**

|  |
| --- |
| class Universiy\_courses:      course\_room = {}      course\_instructor = {}      course\_meeting = {}      course\_names = set()      def \_\_init\_\_(self):          self.load\_courses\_data()      def load\_courses\_data(self):          self.course\_room = {              "CSC101" : 3004,              "CSC102": 4501,              "CSC103": 6755,              "NET110": 1244,              "COM241": 1411          }          self.course\_names.update(self.course\_room.keys())          self.course\_instructor = {              "CSC101" : "Haynes",              "CSC102": "Alvarado",              "CSC103": "Rich",              "NET110": "Burke",              "COM241": "Lee"          }          self.course\_names.update(self.course\_instructor.keys())          self.course\_meeting = {              "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."          }            self.course\_names.update(self.course\_meeting.keys())  if \_\_name\_\_ == "\_\_main\_\_":      university = Universiy\_courses()        university.load\_courses\_data()      print("University course details are loaded")        university\_courses = " , ".join( e for e in university.course\_names)      print(f"Available Courses in the University: {university\_courses}")        while True:          user\_input = str(input("Enter the course name (eg: CSC101, CSC102 ..), else 'q' to quit: ")).strip()          if user\_input == "q":              exit(1)            course\_names = (course.strip().upper() for course in user\_input.split(","))            for course\_name in course\_names:              if (course\_name not in university.course\_names):                  print(f"\n{course\_name} not available in the University")              else:                  print(f"\ncourseName: {course\_name}")                  print("----------------")                  print(f" RoomNumber: {university.course\_room.get(course\_name)}\n InstructorName: {university.course\_instructor.get(course\_name)}\n MeetingTime:{university.course\_meeting.get(course\_name)}") |

**Code execution:**

**A computer screen shot of a program

AI-generated content may be incorrect.**