**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
3. Program terminates when user enters ‘q’ [quit] option

**Code:**

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

**Code execution:**

**A computer screen shot of a program

AI-generated content may be incorrect.**