SRINIVASA EDUCATIONAL SOCIETY'S

PACE INSTITUTE OF TECHNOLOGY & SCIENCES

(AUTONOMOUS)

Approved by AICTE, UGC, New Delhi & Govt. of Andhra Pradesh | Permanently Affiliated to JNTUK, Kakinada, A.P.

ACCREDITED BY **NAAC** WITH **'A'** GRADE | ACCREDITED BY **NBA**

An ISO 9001 : 2008 Certified Institution |'A' Grade Engineering College by Government of A.P.

NH-16, Near Valluramma Temple, ONGOLE - 523 272, A.P., Contact No.: 08592 278315, 9581456310 | www.pace.ac.in

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
(2023-2024)

Internship and mini project based on python programming with Data Engineer

Project title: Course navigator pro

In accordance with requirement of degree of
BACHELOR OF TECHNOLOGY
In
ELECTRICAL AND ELECTRONICS ENGINEERING

Submitted by:
Y.RAHITHYA
21KQ1A0229

Date: 11-06-2024

PROJECT TITLE

COURSE NAVIGATOR PRO

Abstract: The aim of course navigator pro users to manage and query information about various courses. It allows adding course details, querying course fees, available seats, specific course details, checking course existence, displaying enrollment details, counting total filled seats, identifying courses with fees greater than $1000, counting total courses, identifying courses with fees less than $1000, and calculating the total number of seats across all courses.

Description: This Python script serves as a basic course management system. It allows users to input details for multiple courses, including name, fee, enrolled students, total seats, and duration. Upon input, it displays a table of course details, including the name, fee, enrolled students, seats left, and duration. The script provides various functionalities for querying course information, such as querying course fees, available seats, specific course details, checking course existence, displaying enrollment details, counting total filled seats, identifying courses with fees greater than or less than $1000, counting total courses, and calculating the total number of seats across all courses. Users interact with the program through a menu-driven interface, where they can choose options to perform different tasks related to course management. Overall, this script provides essential features for managing and querying information about courses, making it useful for educational institutions, training centers, or any organization offering courses.

Requirements:

1. Functional Requirements:

Input Course Details: Users should be able to input details for multiple courses, including course name, fee, number of enrolled students, total seats, and duration.

Display Course Details: The system should display a formatted table of course details, including the course name, fee, number of enrolled students, seats left, and duration.

Query Course Fee: Users should be able to query the fee for a specific course by providing its name.

Query Available Seats: The system should provide enrollment details for each course, including the number of enrolled students and available seats.

Display Specific Course Details: Users should be able to view specific details of a course by providing its name.

Check Course Existence: Users should be able to check if a course exists in the system by providing its name.

Query Seats Left: The system should display the number of seats left for each course.

Query Courses Under 50 Seats: Users should be able to identify and count courses with an intake of less than 50 seats.

Query Total Filled Seats: The system should calculate and display the total number of filled seats across all courses.

Query Total Courses: The system should count and display the total number of courses.

Query Total Seats: The system should calculate and display the total number of seats across all courses.

2.Non-Functional Requirements:

Mobile App Support: Develop a mobile application to complement the web-based system, providing users with on-the-go access to course information and management capabilities.

User Authentication: Implement user authentication to ensure that only authorized users can access and modify course data.

Course Editing: Enable authorized users (such as administrators or instructors) to edit course details after they've been initially entered, including modifying fees, total seats, or other relevant information.

Course Registration: Allow students to register for courses directly through the system, updating the enrollment numbers accordingly.

Approach:

1. Understanding the Objective:

->Define the purpose of the Course Navigator Probe, which is to assist users in exploring available courses based on their preferences and requirements.

2. Requirement Analysis:

->Identify the target audience and their needs, such as students, professionals, or career counselors.

->Determine the features and functionalities required for effective course navigation, such as search filters, recommendations, and course comparison.

3. Design:

->Design the user interface to be intuitive and user-friendly, allowing users to easily navigate through available courses.

-> Determine the data sources for course information, including course databases, educational institutions' websites, and online learning platforms.

-> Plan the architecture of the navigator probe, considering factors such as scalability, performance, and data retrieval mechanisms.

4. Data Acquisition and Integration:

-> Develop mechanisms to collect and integrate course data from various sources into a centralized repository.

-> Implement data preprocessing techniques to clean, standardize, and organize the course information for efficient navigation.

5. Feature Implementation:

-> Develop search functionalities to allow users to find courses based on criteria such as subject, level, duration, and location.

-> Implement filtering options to refine search results according to user preferences, such as price range, accreditation, and course format.

-> Incorporate recommendation algorithms to suggest relevant courses based on user profiles, browsing history, and similarity to preferred courses.

-> Create course comparison tools to enable users to compare key attributes of multiple courses side by side, facilitating informed decision-making.

6. User Interaction:

-> Design interactive features to engage users and provide personalized experiences, such as user profiles, saved searches, and notifications.

-> Implement feedback mechanisms to gather user input and improve the navigator probe's effectiveness over time.

7. Testing:

-> Conduct extensive testing to ensure the navigator probe's functionality, usability, and performance.

-> Perform usability testing with target users to gather feedback and identify areas for improvement.

-> Validate the accuracy of search results, recommendations, and course comparisons against known benchmarks and user expectations.

8. Deployment:

-> Prepare the Course Navigator Probe for deployment on the desired platform, such as a web application, mobile app, or standalone software.

-> Ensure compatibility with different devices and operating systems to maximize accessibility for users.

-> Provide documentation and support resources to guide users in using the navigator probe effectively.

9. Evaluation and Iteration:

-> Monitor user interactions and feedback to evaluate the navigator probe's effectiveness in meeting users' needs and goals.

- >Analyze usage metrics, such as search patterns, click-through rates, and user satisfaction scores, to identify areas for improvement.

-> Iterate on the navigator probe based on evaluation results, incorporating new features, enhancing existing functionalities, and addressing user feedback to continuously enhance the user experience.

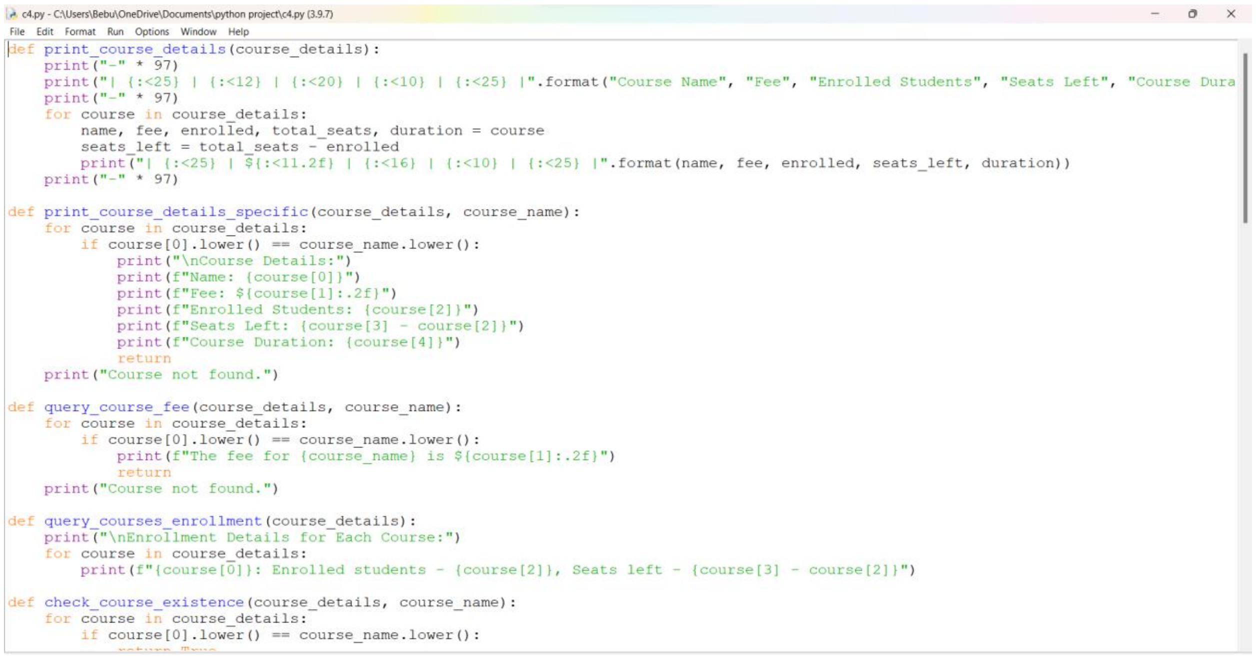
10. Maintenance and Updates:

-> Maintain the Course Navigator Probe by monitoring data sources, updating course information, and addressing technical issues as they arise.

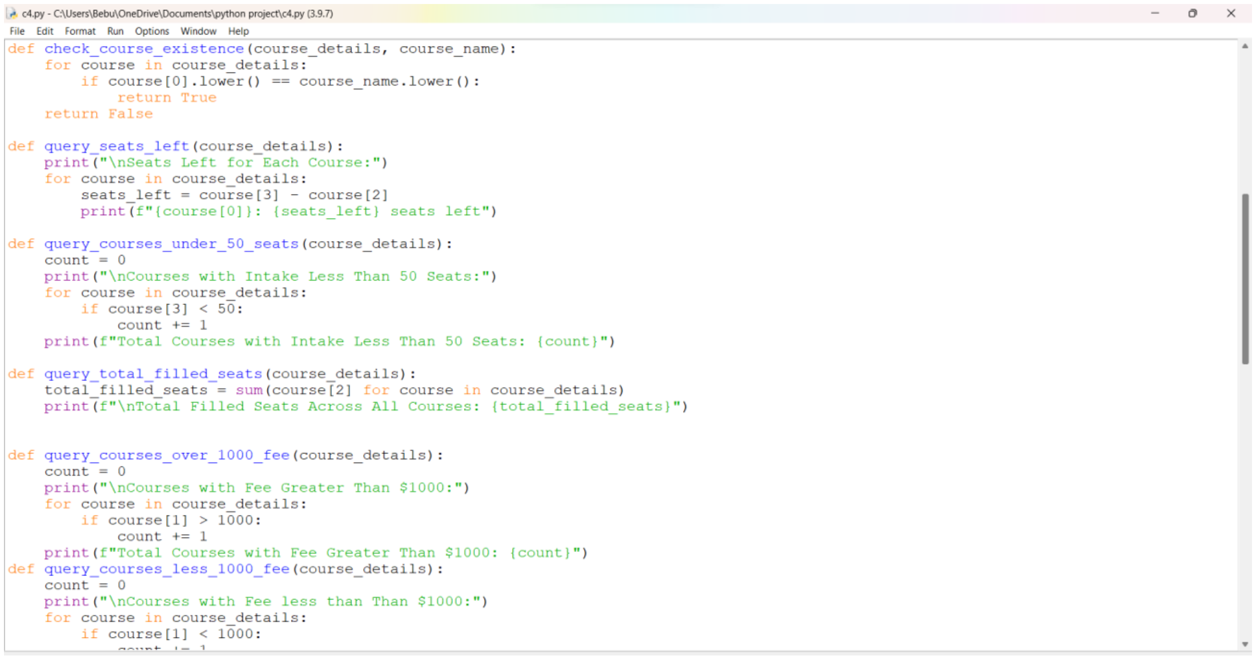
-> Regularly update the navigator probe with new features, course offerings, and improvements to ensure its relevance and usefulness to users.

-> Stay informed about emerging trends and technologies in education and course navigation to incorporate cutting-edge features and capabilities into the navigator probe.

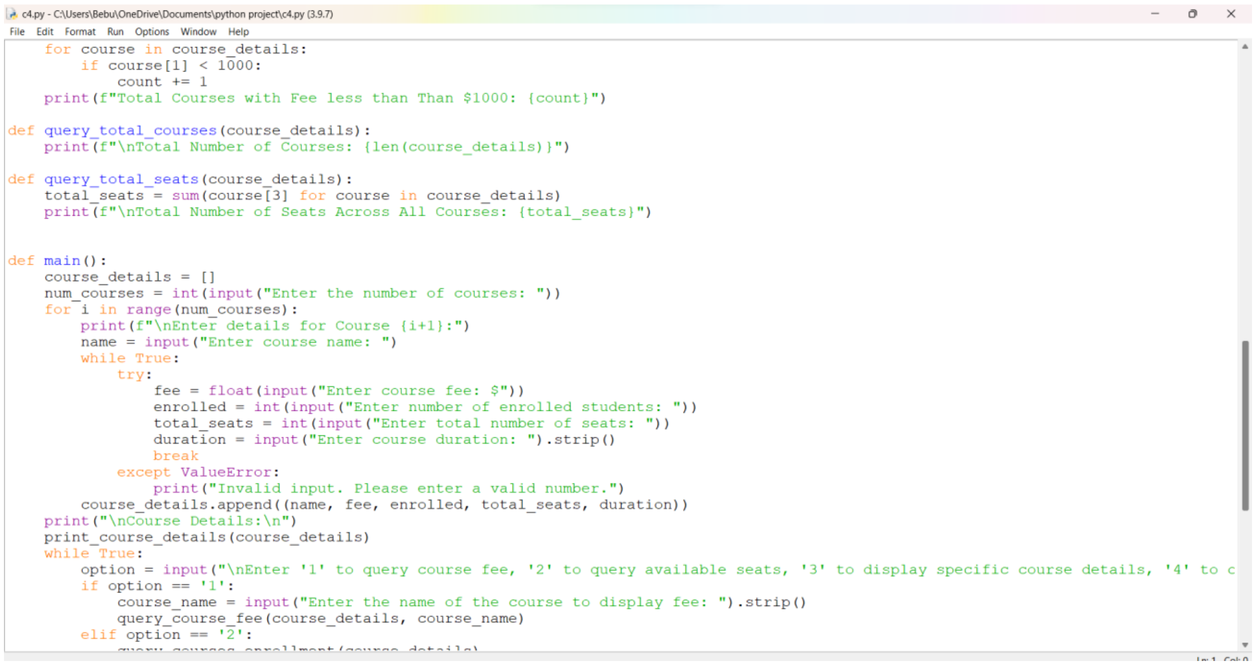
PROGRAM:


c4.py - C:\Users\Bebu\OneDrive\Documents\python project\c4.py (3.9.7)
File Edit Format Run Options Window Help

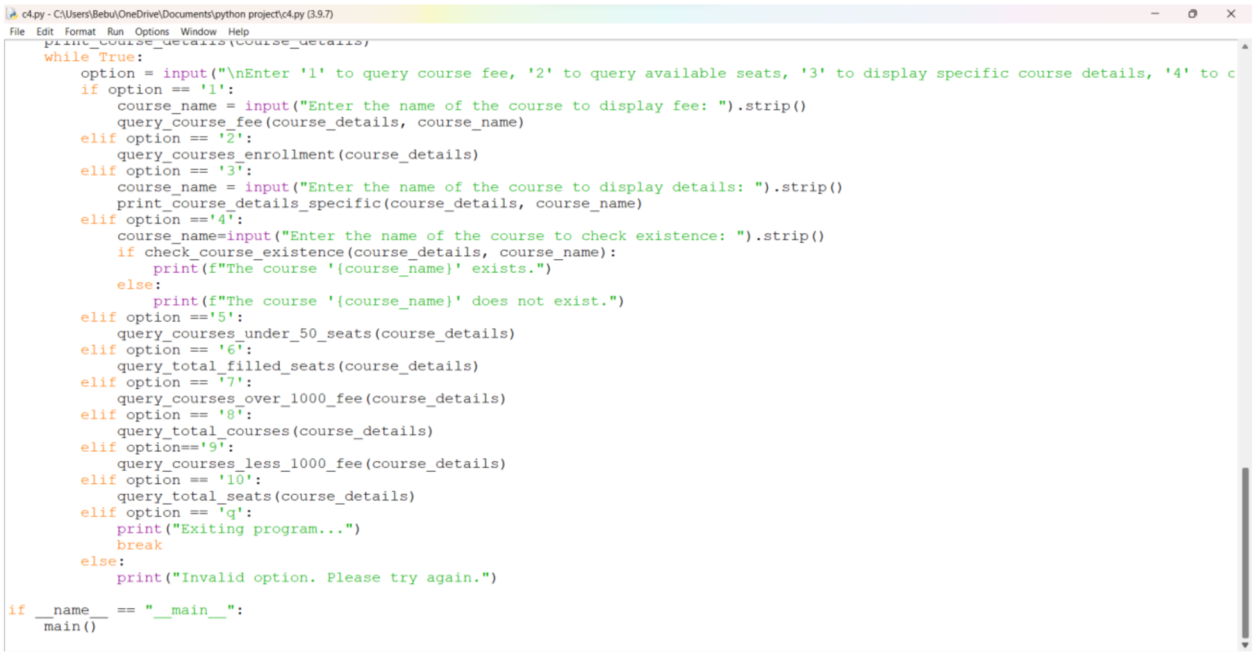
```python
def print_course_details(course_details):
    print("-" * 97)
    print("| {:<25} | {:<12} | {:<20} | {:<10} | {:<25} |".format("Course Name", "Fee", "Enrolled Students", "Seats Left", "Course Dura
    print("-" * 97)
    for course in course_details:
        name, fee, enrolled, total_seats, duration = course
        seats_left = total_seats - enrolled
        print("| {:<25} | ${:<11.2f} | {:<16} | {:<10} | {:<25} |".format(name, fee, enrolled, seats_left, duration))
    print("-" * 97)

def print_course_details_specific(course_details, course_name):
    for course in course_details:
        if course[0].lower() == course_name.lower():
            print("\nCourse Details:")
            print(f"Name: {course[0]}")
            print(f"Fee: ${course[1]:.2f}")
            print(f"Enrolled Students: {course[2]}")
            print(f"Seats Left: {course[3] - course[2]}")
            print(f"Course Duration: {course[4]}")
            return
    print("Course not found.")

def query_course_fee(course_details, course_name):
    for course in course_details:
        if course[0].lower() == course_name.lower():
            print(f"The fee for {course_name} is ${course[1]:.2f}")
            return
    print("Course not found.")

def query_courses_enrollment(course_details):
    print("\nEnrollment Details for Each Course:")
    for course in course_details:
        print(f"{course[0]}: Enrolled students - {course[2]}, Seats left - {course[3] - course[2]}")

def check_course_existence(course_details, course_name):
    for course in course_details:
        if course[0].lower() == course_name.lower():
```


c4.py - C:\Users\Bebu\OneDrive\Documents\python project\c4.py (3.9.7)
File Edit Format Run Options Window Help

```python
def check_course_existence(course_details, course_name):
    for course in course_details:
        if course[0].lower() == course_name.lower():
            return True
    return False

def query_seats_left(course_details):
    print("\nSeats Left for Each Course:")
    for course in course_details:
        seats_left = course[3] - course[2]
        print(f"{course[0]}: {seats_left} seats left")

def query_courses_under_50_seats(course_details):
    count = 0
    print("\nCourses with Intake Less Than 50 Seats:")
    for course in course_details:
        if course[3] < 50:
            count += 1
    print(f"Total Courses with Intake Less Than 50 Seats: {count}")

def query_total_filled_seats(course_details):
    total_filled_seats = sum(course[2] for course in course_details)
    print(f"\nTotal Filled Seats Across All Courses: {total_filled_seats}")


def query_courses_over_1000_fee(course_details):
    count = 0
    print("\nCourses with Fee Greater Than $1000:")
    for course in course_details:
        if course[1] > 1000:
            count += 1
    print(f"Total Courses with Fee Greater Than $1000: {count}")
def query_courses_less_1000_fee(course_details):
    count = 0
    print("\nCourses with Fee less than Than $1000:")
    for course in course_details:
        if course[1] < 1000:
```

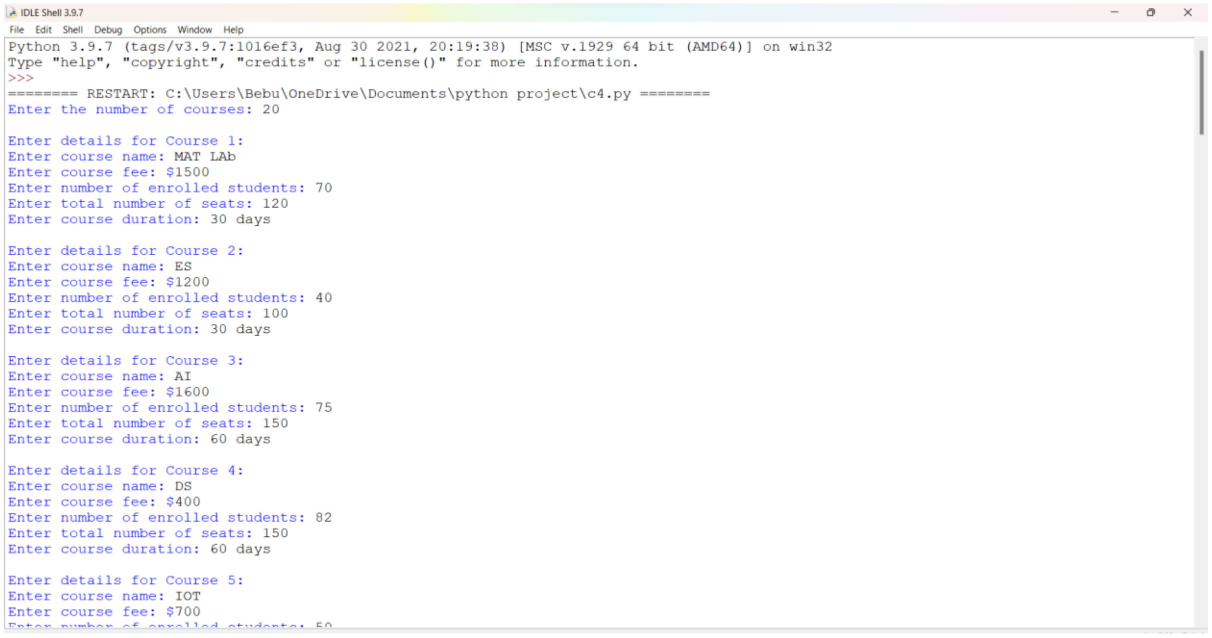

c4.py - C:\Users\Bebu\OneDrive\Documents\python project\c4.py (3.9.7)
File Edit Format Run Options Window Help

```python
    for course in course_details:
        if course[1] < 1000:
            count += 1
    print(f"Total Courses with Fee less than Than $1000: {count}")

def query_total_courses(course_details):
    print(f"\nTotal Number of Courses: {len(course_details)}")

def query_total_seats(course_details):
    total_seats = sum(course[3] for course in course_details)
    print(f"\nTotal Number of Seats Across All Courses: {total_seats}")


def main():
    course_details = []
    num_courses = int(input("Enter the number of courses: "))
    for i in range(num_courses):
        print(f"\nEnter details for Course {i+1}:")
        name = input("Enter course name: ")
        while True:
            try:
                fee = float(input("Enter course fee: $"))
                enrolled = int(input("Enter number of enrolled students: "))
                total_seats = int(input("Enter total number of seats: "))
                duration = input("Enter course duration: ").strip()
                break
            except ValueError:
                print("Invalid input. Please enter a valid number.")
        course_details.append((name, fee, enrolled, total_seats, duration))
    print("\nCourse Details:\n")
    print_course_details(course_details)
    while True:
        option = input("\nEnter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to c
        if option == '1':
            course_name = input("Enter the name of the course to display fee: ").strip()
            query_course_fee(course_details, course_name)
        elif option == '2':
            query_courses_enrollment(course_details)
```

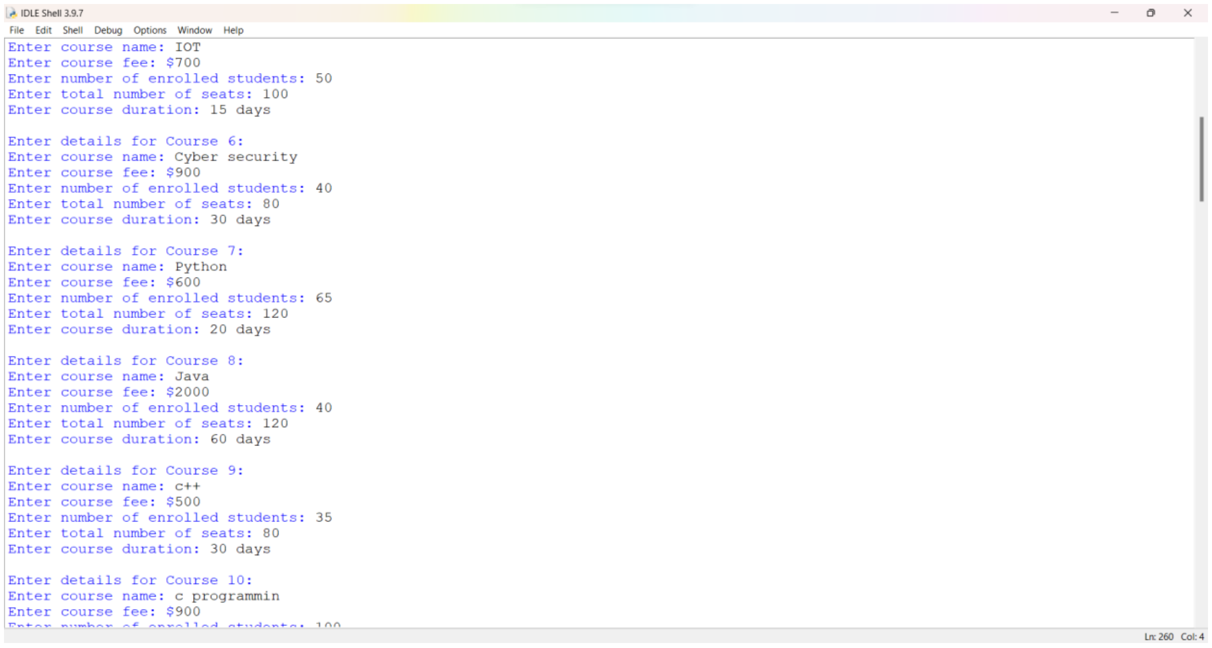

c4.py - C:\Users\Bebu\OneDrive\Documents\python project\c4.py (3.9.7)
File Edit Format Run Options Window Help

```python
    print_course_details(course_details)
    while True:
        option = input("\nEnter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to c
        if option == '1':
            course_name = input("Enter the name of the course to display fee: ").strip()
            query_course_fee(course_details, course_name)
        elif option == '2':
            query_courses_enrollment(course_details)
        elif option == '3':
            course_name = input("Enter the name of the course to display details: ").strip()
            print_course_details_specific(course_details, course_name)
        elif option =='4':
            course_name=input("Enter the name of the course to check existence: ").strip()
            if check_course_existence(course_details, course_name):
                print(f"The course '{course_name}' exists.")
            else:
                print(f"The course '{course_name}' does not exist.")
        elif option =='5':
            query_courses_under_50_seats(course_details)
        elif option == '6':
            query_total_filled_seats(course_details)
        elif option == '7':
            query_courses_over_1000_fee(course_details)
        elif option == '8':
            query_total_courses(course_details)
        elif option=='9':
            query_courses_less_1000_fee(course_details)
        elif option == '10':
            query_total_seats(course_details)
        elif option == 'q':
            print("Exiting program...")
            break
        else:
            print("Invalid option. Please try again.")

if __name__ == "__main__":
    main()
```

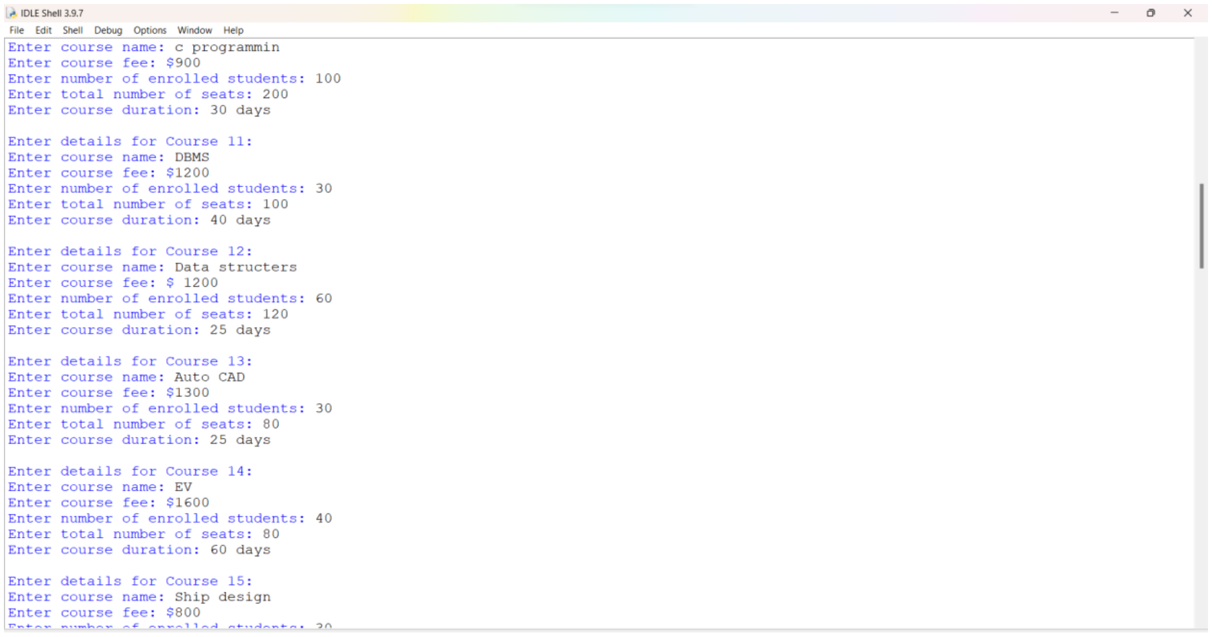
OUTPUT:



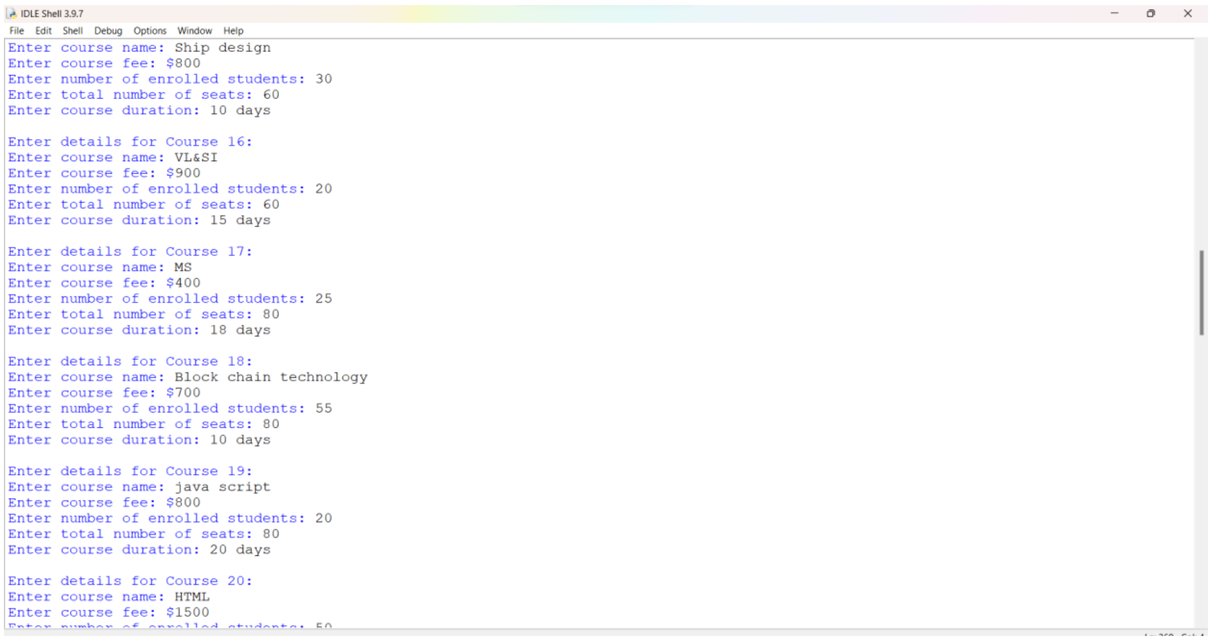
```
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:\Users\Bebu\OneDrive\Documents\python project\c4.py ========
Enter the number of courses: 20

Enter details for Course 1:
Enter course name: MAT LAb
Enter course fee: $1500
Enter number of enrolled students: 70
Enter total number of seats: 120
Enter course duration: 30 days

Enter details for Course 2:
Enter course name: ES
Enter course fee: $1200
Enter number of enrolled students: 40
Enter total number of seats: 100
Enter course duration: 30 days

Enter details for Course 3:
Enter course name: AI
Enter course fee: $1600
Enter number of enrolled students: 75
Enter total number of seats: 150
Enter course duration: 60 days

Enter details for Course 4:
Enter course name: DS
Enter course fee: $400
Enter number of enrolled students: 82
Enter total number of seats: 150
Enter course duration: 60 days

Enter details for Course 5:
Enter course name: IOT
Enter course fee: $700
```



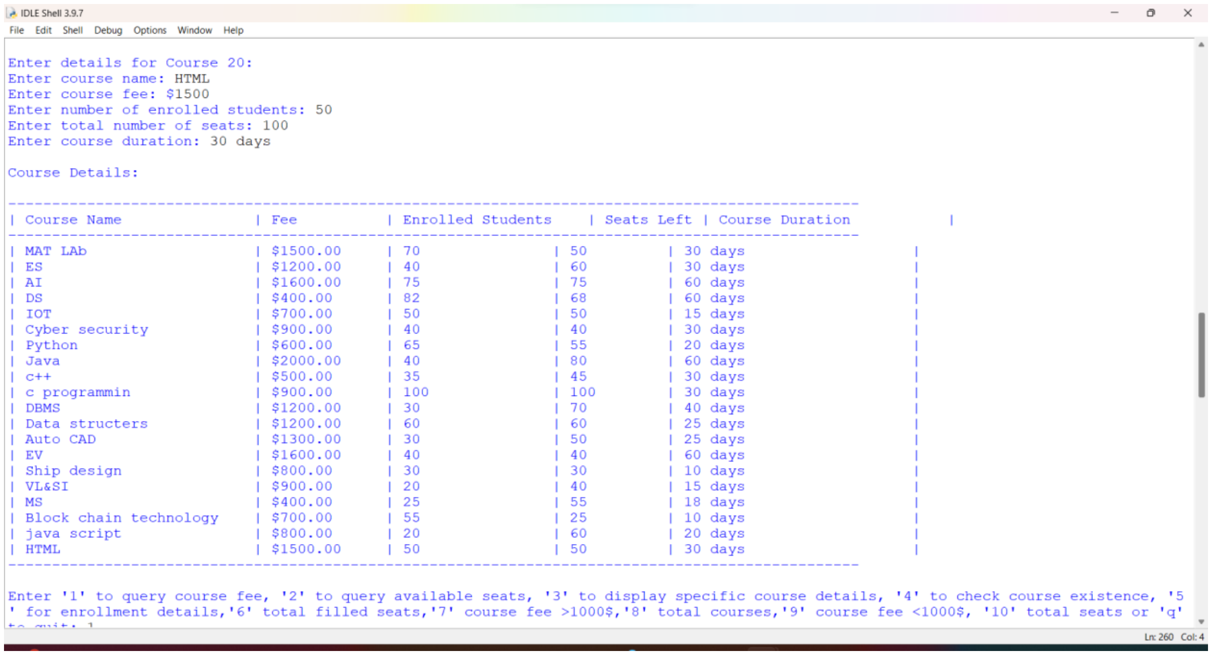
```
Enter course name: IOT
Enter course fee: $700
Enter number of enrolled students: 50
Enter total number of seats: 100
Enter course duration: 15 days

Enter details for Course 6:
Enter course name: Cyber security
Enter course fee: $900
Enter number of enrolled students: 40
Enter total number of seats: 80
Enter course duration: 30 days

Enter details for Course 7:
Enter course name: Python
Enter course fee: $600
Enter number of enrolled students: 65
Enter total number of seats: 120
Enter course duration: 20 days

Enter details for Course 8:
Enter course name: Java
Enter course fee: $2000
Enter number of enrolled students: 40
Enter total number of seats: 120
Enter course duration: 60 days

Enter details for Course 9:
Enter course name: c++
Enter course fee: $500
Enter number of enrolled students: 35
Enter total number of seats: 80
Enter course duration: 30 days

Enter details for Course 10:
Enter course name: c programmin
Enter course fee: $900
```



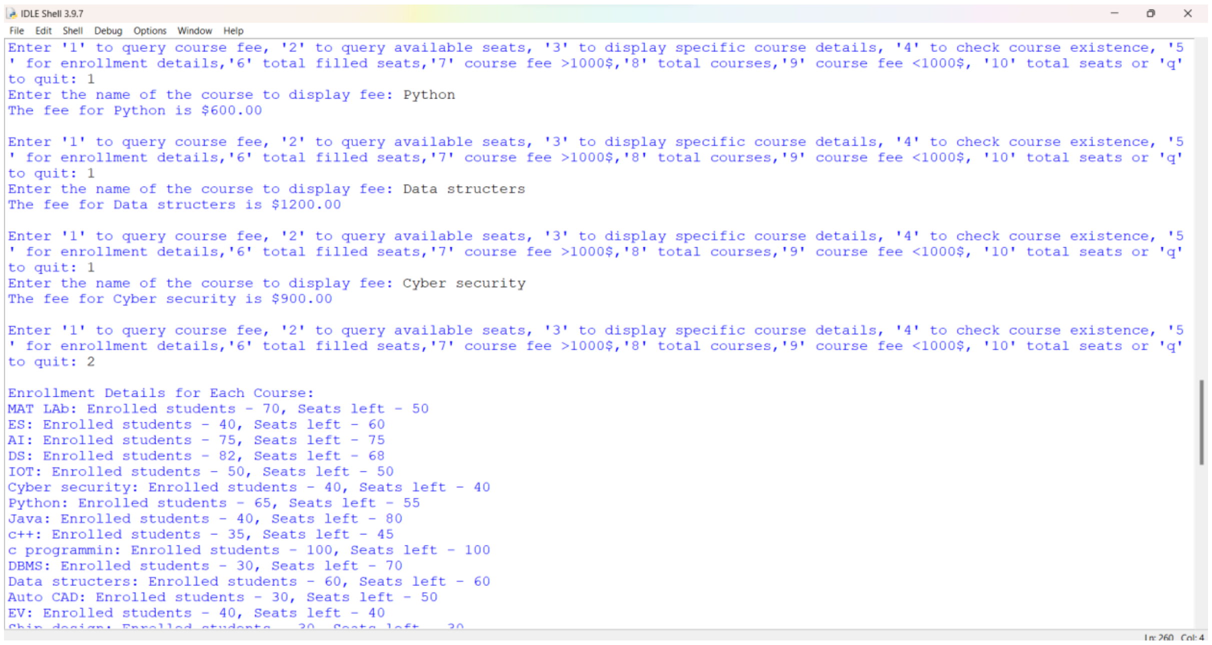
```
Enter course name: c programmin
Enter course fee: $900
Enter number of enrolled students: 100
Enter total number of seats: 200
Enter course duration: 30 days

Enter details for Course 11:
Enter course name: DBMS
Enter course fee: $1200
Enter number of enrolled students: 30
Enter total number of seats: 100
Enter course duration: 40 days

Enter details for Course 12:
Enter course name: Data structers
Enter course fee: $ 1200
Enter number of enrolled students: 60
Enter total number of seats: 120
Enter course duration: 25 days

Enter details for Course 13:
Enter course name: Auto CAD
Enter course fee: $1300
Enter number of enrolled students: 30
Enter total number of seats: 80
Enter course duration: 25 days

Enter details for Course 14:
Enter course name: EV
Enter course fee: $1600
Enter number of enrolled students: 40
Enter total number of seats: 80
Enter course duration: 60 days

Enter details for Course 15:
Enter course name: Ship design
Enter course fee: $800
Enter number of enrolled students: 30
```



```
Enter course name: Ship design
Enter course fee: $800
Enter number of enrolled students: 30
Enter total number of seats: 60
Enter course duration: 10 days

Enter details for Course 16:
Enter course name: VL&SI
Enter course fee: $900
Enter number of enrolled students: 20
Enter total number of seats: 60
Enter course duration: 15 days

Enter details for Course 17:
Enter course name: MS
Enter course fee: $400
Enter number of enrolled students: 25
Enter total number of seats: 80
Enter course duration: 18 days

Enter details for Course 18:
Enter course name: Block chain technology
Enter course fee: $700
Enter number of enrolled students: 55
Enter total number of seats: 80
Enter course duration: 10 days

Enter details for Course 19:
Enter course name: java script
Enter course fee: $800
Enter number of enrolled students: 20
Enter total number of seats: 80
Enter course duration: 20 days

Enter details for Course 20:
Enter course name: HTML
Enter course fee: $1500
Enter number of enrolled students: 50
```



```
Enter details for Course 20:
Enter course name: HTML
Enter course fee: $1500
Enter number of enrolled students: 50
Enter total number of seats: 100
Enter course duration: 30 days

Course Details:

-------------------------------------------------------------------------------------------------
| Course Name                 | Fee           | Enrolled Students    | Seats Left | Course Duration           |
-------------------------------------------------------------------------------------------------
| MAT LAb                     | $1500.00      | 70                  | 50          | 30 days                   |
| ES                          | $1200.00      | 40                  | 60          | 30 days                   |
| AI                          | $1600.00      | 75                  | 75          | 60 days                   |
| DS                          | $400.00       | 82                  | 68          | 60 days                   |
| IOT                         | $700.00       | 50                  | 50          | 15 days                   |
| Cyber security              | $900.00       | 40                  | 40          | 30 days                   |
| Python                      | $600.00       | 65                  | 55          | 20 days                   |
| Java                        | $2000.00      | 40                  | 80          | 60 days                   |
| c++                         | $500.00       | 35                  | 45          | 30 days                   |
| c programmin                | $900.00       | 100                 | 100         | 30 days                   |
| DBMS                        | $1200.00      | 30                  | 70          | 40 days                   |
| Data structers              | $1200.00      | 60                  | 60          | 25 days                   |
| Auto CAD                    | $1300.00      | 30                  | 50          | 25 days                   |
| EV                          | $1600.00      | 40                  | 40          | 60 days                   |
| Ship design                 | $800.00       | 30                  | 30          | 10 days                   |
| VL&SI                       | $900.00       | 20                  | 40          | 15 days                   |
| MS                          | $400.00       | 25                  | 55          | 18 days                   |
| Block chain technology      | $700.00       | 55                  | 25          | 10 days                   |
| java script                 | $800.00       | 20                  | 60          | 20 days                   |
| HTML                        | $1500.00      | 50                  | 50          | 30 days                   |
-------------------------------------------------------------------------------------------------

Enter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to check course existence, '5
' for enrollment details,'6' total filled seats,'7' course fee >1000$,'8' total courses,'9' course fee <1000$, '10' total seats or 'q'
to quit: 1
```



```
Enter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to check course existence, '5
' for enrollment details,'6' total filled seats,'7' course fee >1000$,'8' total courses,'9' course fee <1000$, '10' total seats or 'q'
to quit: 1
Enter the name of the course to display fee: Python
The fee for Python is $600.00

Enter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to check course existence, '5
' for enrollment details,'6' total filled seats,'7' course fee >1000$,'8' total courses,'9' course fee <1000$, '10' total seats or 'q'
to quit: 1
Enter the name of the course to display fee: Data structers
The fee for Data structers is $1200.00

Enter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to check course existence, '5
' for enrollment details,'6' total filled seats,'7' course fee >1000$,'8' total courses,'9' course fee <1000$, '10' total seats or 'q'
to quit: 1
Enter the name of the course to display fee: Cyber security
The fee for Cyber security is $900.00

Enter '1' to query course fee, '2' to query available seats, '3' to display specific course details, '4' to check course existence, '5
' for enrollment details,'6' total filled seats,'7' course fee >1000$,'8' total courses,'9' course fee <1000$, '10' total seats or 'q'
to quit: 2

Enrollment Details for Each Course:
MAT LAb: Enrolled students - 70, Seats left - 50
ES: Enrolled students - 40, Seats left - 60
AI: Enrolled students - 75, Seats left - 75
DS: Enrolled students - 82, Seats left - 68
IOT: Enrolled students - 50, Seats left - 50
Cyber security: Enrolled students - 40, Seats left - 40
Python: Enrolled students - 65, Seats left - 55
Java: Enrolled students - 40, Seats left - 80
c++: Enrolled students - 35, Seats left - 45
c programmin: Enrolled students - 100, Seats left - 100
DBMS: Enrolled students - 30, Seats left - 70
Data structers: Enrolled students - 60, Seats left - 60
Auto CAD: Enrolled students - 30, Seats left - 50
EV: Enrolled students - 40, Seats left - 40
Ship design: Enrolled students - 30, Seats left - 30
```

Explanation: This Python script provides a command-line interface for managing course details, enrollment, and fee querying. Here's a brief explanation of how it works:

1Course Details Input: The script prompts the user to enter details for multiple courses, including name, fee, enrolled students, total seats, and duration..

2. Print Course Details: The print_course_details function prints a tabular representation of all course details, including the course name, fee, number of enrolled students, seats left, and duration.

3. Querying Functions:

- **query_course_fee**: Allows users to query the fee for a specific course by providing its name.
- **query_courses_enrollment**: Displays enrollment details for each course, including the number of enrolled students and available seats.
- **print_course_details_specific**: Displays detailed information about a specific course based on its name.
- **check_course_existence**: Checks if a course exists based on the provided name.
- **query_seats_left**: Prints the number of seats left for each course.
- **query_courses_under_50_seats**: Prints the count of courses with an intake of less than 50 seats.
- **query_total_filled_seats**: Prints the total number of filled seats across all courses.
- **query_courses_over_1000_fee**: Prints the count of courses with a fee greater than $1000.
- **query_courses_less_1000_fee**: Prints the count of courses with a fee less than $1000.
- **query_total_courses**: Prints the total number of courses.
- **query_total_seats**: Prints the total number of seats across all courses.

4. Main Function: The main function serves as the entry point of the script. It presents a menu-driven interface to perform various operations:

- Query course fee
- Query available seats
- Display specific course details
- Check course existence
- View enrollment details
- Calculate total filled seats
- Count courses with fee greater than $1000
- Count courses with fee less than $1000
- View total number of courses
- View total number of seats
- Quit the program

5 Error Handling: The script handles errors gracefully, informing the user if invalid input is provided and prompting them to try again.

Conclusion: In conclusion, the provided Python script offers a comprehensive course management system through a command-line interface. It allows users to input details for multiple courses, view various statistics, and perform queries based on specific criteria. Here's a summary of its features.

Functionality:

The script provides options to query course fee, available seats, specific course details, check course existence, enrollment details, and more.

Flexibility:

- Users can interact with the system through a menu-driven interface, selecting options based on their needs.
- Error handling is implemented to ensure robustness, prompting users to re-enter data in case of invalid input.

Readability:

- The code is well-structured and easy to understand, with clear function names and comments where necessary.
- The use of functions enhances modularity and reusability, making it easier to maintain and extend the codebase.

Overall, this script provides a practical solution for managing course-related information, suitable for educational institutions or training centers looking for a simple yet effective way to track and analyze course data.