

Assessment

Module 13) Python - Fundamentals of Python Language

Module 14) Python - Collections, Functions, and Modules in Python

Case Overview

You are a trainee Python developer at BrightFuture Academy, a coaching institute for school and college students.

Currently, student attendance is managed using paper registers and word-of-mouth updates, leading to missed records, lack of tracking, and student disputes.

The academy has requested a Python-based console program, named EduTrack, that allows staff to manage daily student attendance, generate basic attendance reports, and track defaulters -- using only core Python programming features.

Core Functionality

Attendance Marking • Allows staff to record daily attendance for each student.

- Captures student name, roll number, course, and date.

Attendance Report • Generates an attendance summary for a given student or full class.

- Calculates total days present/absent and flags defaulters (attendance < 75%).

Key Competencies Tested

Python Concepts:

- Functions
- Looping (for, while)
- Collections (lists, dictionaries)
- Input/output formatting

Practical Considerations:

- In-memory data handling for session-level testing
- Function-based, clean modular design
- Input validation (e.g., valid dates, no duplicate entries)
- Report generation with simple tabular formatting

Reflective Thinking

- How can this be extended to save daily logs using file handling?
- Could a future version support teacher login and per-course attendance?
- Can this system be integrated with SMS/email alerts for absentee students?

CODE:

```
import datetime
```

```
students = []
```

```
attendance_records = {}
```

```
def add_student():
```

```
    name = input("enter student name: ")
```

```
    roll = input("enter roll number: ")
```

```
    course = input("enter course: ")
```

```
    students.append({"name": name, "roll": roll, "course": course})
```

```
    attendance_records[roll] = []
```

```
def mark_attendance():
```

```
    date = input("enter date (yyyy-mm-dd) or leave blank for today: ")
```

```
    if not date:
```

```
date = str(datetime.date.today())
```

```
for student in students:
```

```
    status = input(f"is {student['name']} (roll: {student['roll']}) present? (y/n): ").lower()
```

```
    record = {"date": date, "status": True if status == 'y' else False}
```

```
    attendance_records[student["roll"]].append(record)
```

```
def generate_report():
```

```
    print("\n--- edutrack attendance report ---")
```

```
    print(f"{'roll':<10} {'name':<15} {'present':<8} {'absent':<8} {'%':<8} {'status':<10}")
```

```
for student in students:
```

```
    roll = student["roll"]
```

```
    records = attendance_records[roll]
```

```
    total = len(records)
```

```
    if total == 0:
```

```
        print(f"{'roll':<10} {student['name']:<15} no records")
```

```
        continue
```

```
    present = sum(1 for r in records if r["status"])
```

```
    absent = total - present
```

```
    percentage = (present / total) * 100
```

```
    defaulter = "low" if percentage < 75 else "ok"
```

```
print(f"{roll:<10} {student['name']:<15} {present:<8} {absent:<8} {percentage:<8.2f}  
{default:<10}")
```

```
def main():
```

```
    while True:
```

```
        print("\n1. add student\n2. mark attendance\n3. generate report\n4. exit")
```

```
        choice = input("select option: ")
```

```
        if choice == "1":
```

```
            add_student()
```

```
        elif choice == "2":
```

```
            if not students:
```

```
                print("add students first")
```

```
            else:
```

```
                mark_attendance()
```

```
        elif choice == "3":
```

```
            generate_report()
```

```
        elif choice == "4":
```

```
            break
```

```
        else:
```

```
            print("invalid choice")
```

```
if __name__ == "__main__":
```

```
    main()
```

OUTPUT:

1. add student
2. mark attendance
3. generate report
4. exit

select option: 1

enter student name: yagnesh

enter roll number: 21

enter course: cse ai

1. add student
2. mark attendance
3. generate report
4. exit

select option: 2

enter date (yyyy-mm-dd) or leave blank for today:

is yagnesh (roll: 21) present? (y/n): y

1. add student
2. mark attendance
3. generate report
4. exit

select option: 3

--- edutrack attendance report ---

roll	name	present	absent	%	status
21	yagnesh	1	0	100.00	ok

1. add student

2. mark attendance

3. generate report

4. exit

select option: 4

=== Code Execution Successful ===