



	School of Engineering & Technology	
	Department: SOET	Session: 2025-26
	Program: MCA (AI & ML)	Semester: I
	Course Code: ETCCPP171	Number of students:
	Course Name: Programming for Problem Solving Using Python	Faculty: Ms. Neha Kaushik

Assignment Number: 01

Submission Instructions:

1. **Submission Deadline:** Assignments must be submitted within one week of the assignment's release date.
2. **Submission Platform:** All assignments are to be submitted via the Learning Management System (LMS) or Moodle (<https://lms.krmangalam.edu.in/>).
3. **GitHub Link:** You must provide a link to your GitHub repository with your submission (Optional).
4. **Individual Submission:** Assignments are to be completed and submitted by each individual student.
5. **Formatting:** All assignments must adhere to the specific format shared in class.

Total Marks: 5

Real-World Problem Context

In schools, colleges, and workplaces, attendance tracking is an essential but time-consuming task. Manual record-keeping can often lead to errors and inefficiency. To simplify this process, this lab assignment aims to build a **Python-based Command-Line Attendance Tracker** that allows users to record student names and timestamps, validate entries, and generate formatted attendance summaries automatically.

This project uses the foundational concepts of Python programming such as lists, dictionaries, control flow, string formatting, and functions — all part of **Unit 1: Python Programming Foundations**.



Learning Objectives

By completing this lab assignment, learners will:

- Practice using **input()**, data type conversion, and basic I/O operations.
- Store and manage attendance records using **lists and dictionaries**.
- Apply **control statements** (`if`, `elif`, `else`) to validate user entries.
- Implement **loops** for multiple record entries.
- Format and display tabular summaries using **f-strings** and string alignment.
- Strengthen **logical and modular programming skills** by creating a real-world attendance system.

Assignment Tasks

Task 1: Setup & Project Initialization

- Create a folder named `attendance_tracker`.
- Inside the folder, create a Python file named `tracker.py`.
- Add a comment header including your **name, date, and assignment title**.
- Print a **welcome message** explaining the purpose of the tool.

Deliverable: `tracker.py` file with setup and project introduction.

Task 2: Input & Data Collection

- Use `input()` to accept the following:
 - Student name
 - Check-in time (e.g., 09:15 AM)
- Store these values in a **dictionary**, where the key is the student's name and the value is the timestamp.
- Ask the user how many entries they want to record and use a **for loop** to collect data accordingly.

Deliverable: Working logic for collecting and storing attendance data.



Task 3: Data Validation

- Add input validation using **conditional statements**:
 - If a student name is left blank, show a message “Name cannot be empty.”
 - If the timestamp is missing, ask the user to re-enter the time.
- Prevent duplicate entries by checking if the student’s name already exists in the dictionary.

Deliverable: Validated attendance entry system using `if...else` conditions.

Task 4: Attendance Summary Generation

- After all entries are made, display a formatted attendance summary using **f-strings**:

```
Student Name      Check-in Time
-----
Riya Sharma      09:00 AM
Arjun Singh      09:10 AM
Meena Verma      09:15 AM
-----
Total Students Present: 3
```

- Use `\t` and `\n` for proper formatting and alignment.

Deliverable: Neatly formatted attendance report displayed on the console.

Task 5: Absentee Validation (Optional Enhancement)

- Ask the user to input the **total number of students** in the class.
- Calculate and print the **number of absentees** by comparing attendance count with total strength.
- Display both “Total Present” and “Total Absent” in the summary.

Deliverable: Extended attendance summary with absentee calculation.

Task 6 (Bonus): Save Attendance Report to File



- Ask the user if they wish to save the attendance record.
- If yes, create a file named `attendance_log.txt` and write:
 - Student names with timestamps
 - Total present and absent count
 - Current date and time of report generation (use `datetime` module).

Deliverable: A `.txt` file containing attendance summary and timestamp.

Submission Instructions

Submit the following items:

- `tracker.py` Python script with all tasks implemented
 - At least **3 sample runs** with different input sets
 - `attendance_log.txt` file (if bonus task completed)
 - Header comments and user-friendly output formatting
 - Submit your **GitHub repository link** (optional) via LMS
-

Evaluation Rubric

Criterion	Weight	Excellent (5)	Good (4)	Fair (3)	Poor (2)
Project setup	10%	Well-structured, reproducible	Functional	Incomplete	Not attempted
Data collection logic	20%	Accurate, clean input handling	Mostly correct	Basic implementation	Incorrect
Data validation	15%	Robust error handling	Partial checks	Limited	Missing
Control & loop logic	20%	Clear and efficient	Partially correct	Repetitive	Incorrect
Output formatting	25%	Clean tabular format	Acceptable	Minimal	Messy
Code quality & readability	10%	Well-documented and modular	Average	Weak	Poor or copied

Academic Integrity & Plagiarism Policy

- All submissions must be **individual and original**.



K.R. MANGALAM UNIVERSITY
THE COMPLETE WORLD OF EDUCATION

- Copying or submitting code written by others will result in **zero marks** and disciplinary action.
- Students can refer to documentation (e.g., Python Docs, tutorials) but must cite any external references in `README.md`.