creativecapsule Logo

Foundation Training '25

CC Programming-Foundations

Instructions

- Write Pseudo code for the below problems
- Test the Pseudo code with the tests provided in the problem statements to validate the logic
- Start coding only after the first two steps are completed.

Overview

Complete all problems.

Task 1 : Save User Input to CSV File

Problem Statement

Ask the user to enter their **name**, **age**, and **email address**.
Use the `string` module to:

- Ensure the name contains only alphabetic characters using `string.ascii_letters`
- Check that the email contains `@` and `.`

Save valid inputs into a file `users.csv` using the `csv` module.
Use `try/except` to:

- Handle invalid input
- Catch file I/O errors

Expected output

Example:

```
Input:
Name: John
Age: 25
Email: john@example.com
```

```
Output:
User saved to users.csv
```

Task 2 : Read from JSON and Display Info

Problem Statement

Given a file `students.json`:

```
[
  {"name": "Alice", "marks": 85},
  {"name": "Bob", "marks": 78}
]
```

Write a program that:

- Reads the file using `json` module
- Displays each student's name and marks
- Use exception handling for FileNotFoundError and JSONDecodeError

Expected output

Output :

```
Alice - 85 marks
Bob - 78 marks
```

Task 3 : Error-Proof Age Entry with Math Validation

Problem Statement

Ask the user to enter their age.
Validate that:

- Age is numeric and between 1 and 120
- Use `math.floor()` to ensure the number is whole

Expected output

Examples:

```
Input: abc
Output: Invalid input. Please enter a numeric age.
```

```
Input: -4
Output: Age must be a positive number.

Input: 22.5
Output: Please enter a whole number.

Input: 25
Output: Age accepted.
```

Task 4 : Reminder App using `datetime`

Problem Statement

Create a reminder app that:

- Accepts a task name and reminder time in `HH:MM`
- Uses `datetime.datetime.now()` to show current date and set reminder

Expected output

Example:

```
Input:
Task: Attend Meeting
Time: 15:45

Output:
Reminder set for 'Attend Meeting' at 15:45 on 2025-06-10
```

Task 5 : CSV Number Reader with Error Handling

Problem Statement

Read a CSV file `numbers.csv` with one column `Number`.

- Use `math.sqrt()` to compute square roots of non-negative numbers
- Skip invalid or negative entries using exception handling
- Store results in `sqrt_results.csv`

Expected output

Sample Input (numbers.csv):

```
Number
25
```

```
-4
hello
16
```

Expected Output (Console):

```
Square root of 25 is 5.0
Skipping invalid or negative entry: -4
Skipping invalid or non-numeric entry: hello
Square root of 16 is 4.0
```

Output File (sqrt_results.csv):

```
Number,SquareRoot
25,5.0
16,4.0
```
