# 

## Practical Project 01

Student Name: David Shaju Menachery

Professor: Stanley Pieda

Due Date: 25-05-2025

**Section 1: Evidence of Learning**

* **Variables**

In record.py, the variables like self.sample\_no, self.product, and self.result are used to store individual data fields from each CSV record. These represent the state of a Record object.

* **Methods**

The method \_\_str\_\_(self) in record.py formats the data for display when printing a record. It is a special method used to define how objects are represented as strings.

* **Loop Structure**

In main.py, the loop for i, row in enumerate(reader): goes through each row of the CSV file one by one. It allows the program to process multiple records efficiently.

* **File I/O**

The line with open(filename, mode='r', encoding='cp1252') as file: in main.py shows how the program opens and reads the CSV file from disk. This is essential for loading data into the program.

* **Exception Handling**

The try and except blocks in load\_records() catch errors like a missing file or a bad column name. This prevents the program from crashing and shows a user-friendly message instead.

* **API Use**

The csv module is a built-in Python API that simplifies reading and parsing CSV files. csv.DictReader is used to automatically map each row to a dictionary using the header row.

* **Data Structure**

A list named records is used to store all the loaded Record objects. This allows for efficient storage, access, and display of multiple rows of data from the CSV.

**Section 2: Program Demonstration via Screenshots**

* **Screenshot 1: Terminal Output of Running the Program**

**A screenshot of a computer

AI-generated content may be incorrect.**

* **Description**

This screenshot shows the program successfully reading the CSV dataset and displaying the first 5 records. It confirms that my name is printed, the file is read correctly, and the output is structured using the Record class.

* **Screenshot 2: VS Code with main.py Open**

A computer screen shot of a program

AI-generated content may be incorrect.

* **Description:**

This screenshot shows the main.py file, which contains the program logic to load the CSV file, handle exceptions, and print the records using a loop. My name is shown at the top in the file docstring.

* **Screenshot 3: VS Code with record.py Open**

A screenshot of a computer program

AI-generated content may be incorrect.

* **Description:**

This file defines the Record class used to represent each dataset row. It demonstrates object-oriented programming and includes a custom \_\_str\_\_ method to format the output. My author info is also included.

* **Screenshot 4: Project Folder View**

A screenshot of a computer

AI-generated content may be incorrect.

* **Description**

This screenshot confirms that all required project files are present and organized. It includes the dataset, code files, documentation files like README.md and .gitignore, and the report itself — all in one folder ready for submission.

**Section 3: Source Code Commenting Example**

"""

Course: CST8002 - Programming Language Research Project

Professor: Stanley Pieda

Due Date: May 25, 2025

Author: David Shaju Menachery

"""

class Record:

"""

Represents a data record from the provided dataset.

Each instance stores information such as sample number,

region, function, origin, product type, result value, and unit.

"""

def \_\_init\_\_(self, sample\_no, region, function, origin, product, result, unit):

"""

Initializes a Record object with values from one row of the CSV.

Parameters:

sample\_no (str): Sample number or ID

region (str): Geographic region

function (str): Functional category

origin (str): Origin of the sample

product (str): Product description

result (str): Test result value

unit (str): Unit of measurement

"""

self.sample\_no = sample\_no

self.region = region

self.function = function

self.origin = origin

self.product = product

self.result = result

self.unit = unit

def \_\_str\_\_(self):

"""

Returns a formatted string representation of the Record object.

"""

return (f"{self.sample\_no} | {self.region} | {self.function} | {self.origin} | "

f"{self.product} | {self.result} {self.unit}")