|  |  |
| --- | --- |
| Blue letters on a white background  Description automatically generated | Text  Description automatically generated with medium confidence |

# Student Assessment Submission and Declaration

When submitting evidence for assessment, you must sign a declaration confirming that the work is your own.

|  |  |  |  |
| --- | --- | --- | --- |
| Student name: |  | ESL ID No |  |
| BSU ID No |  |
| Submission date: | | |  |
|  | | |  |
| Programme: |  | | |
| Module name and code: |  | | |
| Title: |  | | |
| Assessor name: |  | | |

**Plagiarism**

Plagiarism is a form of cheating. Plagiarism must be avoided at all costs and students who break the rules, however innocently, may be penalised. It is your responsibility to ensure that you understand correct referencing practices. As a university level student, you are expected to use appropriate references throughout and keep carefully detailed notes of all your sources of materials for material you have used in your work, including any material downloaded from the Internet. Please consult the relevant unit lecturer or your course tutor if you need any further advice.

|  |  |  |  |
| --- | --- | --- | --- |
| **Student declaration**  I certify that the assignment submission is entirely my own work. I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
| Student signature: |  | Date: |  |

Development Document: CSV-Based Data Processing Program

# Table of Contents

Contents

[Table of Contents 1](#_Toc198998897)

[Introduction 2](#_Toc198998898)

[How to Run the Program 2](#_Toc198998899)

[Solution Design 2](#_Toc198998900)

[Reflective Evaluation 3](#_Toc198998901)

[Appendix: Full Code Listing 4](#_Toc198998902)

# Introduction

The task consisted of writing a Python-based Data Processing Program that would read user data from an input CSV file, treat it, and write results into an output CSV file. The general intent was to demonstrate basic programming skills such as file I/O, handling data, arithmetic operations, and neat output using standard Python libraries. I broke down the requirement into modular and testable pieces. The end result has a command-line executable script, some sample data, and a shell script for your quick and easy testing. The GitHub repository for this project is available here: <https://github.com/example/data-processing-project>

# How to Run the Program

**To execute this code in Visual Studio Code (VS Code):**

1. Open the folder holding process\_data.py and input.csv in VS Code.
2. Open a new terminal within VS Code.
3. Execute the following command:  
   python process\_data.py input.csv
4. The programme will output output.csv in the same directory.

**To execute from a Windows CMD or PowerShell command prompt:**

1. Go to project folder using cd command.
2. Run python process\_data.py input.csv

Ensure Python is installed and has been added to your PATH environment variable. You can check if Python is installed using python --version.

# Solution Design

**Flowchart**

1. Start  
2. Get input file path from command line  
3. Open a CSV file  
4. Iterate through each row and parse out name, age, and height  
5. Calculate future age and height in centimeters  
6. Create greeting message  
7. Add processed data into output list  
8. Output list to output.csv  
9. End

**Script Walkthrough**  
  
The central script, process\_data.py, is organized into three major sections:  
  
Input Handling: The script takes input file name from command line using sys.argv and verifies if arguments and files are present or not  
  
Data Processing: The script reads rows of data into dictionaries using Python's csv.DictWriter. For an individual we have Name, Age, and Height in meters. The following are calculated by the program

**Future Age is Age + 10**  
**Height (cm) = Height (m) × 100**

Greetings Generation: A greeting is generated based upon an f-string that takes into account a person's current age, age in the future, and their metric heights as well. This is saved in a new dictionary under all final output fields.  
  
All the processed data is stored into output.csv utilizing csv.DictWriter along with headers Name, Age, Future Age, Height (m), Height (cm), Greeting

**Technical Highlights**

Validation is done for all required fields and ensures there are proper data types.  
  
Modularity: The code has been divided into separate logical sections, which could be easily modified or expanded.  
  
Compatibility: This script uses no external libraries. This offers compatibility across any system that has Python, without requiring pip installation commands.

# Reflective Evaluation

# This exercise was an excellent way to get a good feel for file I/O operations, error handling, and handling CSV file data in Python. A highlight was constructing human-readable output from a string containing values from several variables. I appreciated challenging myself to present numerical values (e.g., height in cm) in a standard format while being accurate in output CSV as well. One of the important lessons learned was the need for verifying input data. I first faced problems when blank lines or unexpected data types made the program crash. Having good error handling was essential and proved useful in making the final script highly reliable. If I were to continue refining this project, I would add user interactivity—maybe allow users to select which calculations are to be done—or implement a graphical user interface (GUI) through tkinter. Being able to output results in various forms (e.g., JSON, XML) would provide further flexibility for the program. Finally, there's value in understanding object-oriented programming (OOP) and Python modules. Encapsulation of code into functions and classes would facilitate reusability and testability within the codebase. Understanding about automatic test frameworks like unittest or pytest would help in avoiding regressions that could result from future changes.

# 

# Appendix: Full Code Listing

**process\_data.py**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**run\_process.sh**

**A screen shot of a computer

AI-generated content may be incorrect.**