Problem-Solving Activity 2

GCIS 123: Software Development & Problem Solving I

Fall 2025

Total marks: 100 **Weightage: 10% of Course Grades**

Assignment: Python Programming – Control Structures and File Handling

Important Note (Group-Based Assignment)

- This is a group-based assignment. The instructor will create random groups on MyCourses. **2 to 3 students in a group**.
- Each group must include the **names and student IDs** of all members at the top of the file. Failure to provide this information will **impact on the group's overall grade**.
- You must work together with your group members to complete this activity.
- Each student must provide a **manifesto**, either within the code (as comments) or on a separate page. The manifesto should clearly state the **student's contribution** to problem-solving and coding. Failure to include this manifesto may affect the group's overall grade.
- Every member should actively participate in coding, discussion, and testing.
- **Only one student** (chosen by the group) should submit the final code on behalf of the team. There will be presentations in the following classes after the deadline.
  Every group member must present part of the solution. Instructor will choose randomly which group to present, all should be prepared well.
- All group members will get the same grade for the project. However, your **individual marks may change based on your contribution, your manifesto**, and how well you present or answer questions.
- **Absentees will lose presentation marks (zero if not present).**

Goals of the Assignment:

This assignment will test your understanding of:

- Conditionals
- For and While Loops
- String operations
- File handling (reading data from a **CSV file**)
- Incremental development and teamwork

You will solve a **real-world inspired problem** step by step, improving your program in stages.

Scenario: Library Book Borrowing System

The university library needs a simple system to manage borrowed books. You have been asked to write a Python program that processes a **csv file** containing student names and the number of books they borrowed. The program will check borrowing limits, calculate fines, and generate a summary report.

Task 1: Implement check_limit(borrowed) function

- Input: an integer borrowed representing the number of books borrowed.
- Rules:
  - If borrowed ≤ 3 → Return "Within limit"
  - If borrowed > 3 and ≤ 6 → Return "Over limit: Fine $5"
  - If borrowed > 6 → Return "Over limit: Fine $10"
  - If borrowed < 0 → Return "Error: Invalid number of books"

Task 2: Implement process_borrowers(filename) function

- Read a **csv file** where each line has the format:
  Name,BooksBorrowed

Example:

Ali,2
Sara,5
John,-1
Mary,abc
Davi

- For each student:
  - Use a loop to process the file line by line.
  - Call **check_limit()** to determine the status.
  - Print the student's name and their status.
- If Value is not a valid integer, print an error message:
  "Error: Non-numeric value for <Name>

Task 3: Implement calculate_average_books(filename) function

- Read the file again.
- Calculate the average number of books borrowed by all students with valid entries.
- Print the result (rounded to 2 decimal places).

Task 4: Implement count_over_limit(filename) function

- Count how many students borrowed more than 3 books.
- Print the total count.

Task 5: Write main() function

- Prompt the user for a filename.
- Use a while loop to retry until a valid file is provided.
- Call all the above functions:
  1. process_borrowers()
  2. calculate_average_books()
  3. count_over_limit()

Sample File Content (borrowers.csv)

Ali,2
Sara,5
John,-1

```
Mary,abc
David,7
```

Expected Output

```
Ali: Within limit
Sara: Over limit: Fine $5
Error: Invalid number of books for John
Error: Non-numeric value for Mary
David: Over limit: Fine $10

Average number of books borrowed: 4.67
Number of students over the limit: 2
```

Grading Rubric

Criteria	Points
Correct implementation of 4 major functions	40
Correct file handling, calculations, and validation	20
Well-structured and readable code, with docstrings	10
Output formatting and clarity	10
Group collaboration + presentation (all members)	20
Any code copied from AI tools, or using uncovered concepts will be a penalty	
Activity without presentation will grade the submission as 0	

Submission Instructions

- Submit a single Python script (library_system.py) containing all functions.
- Add inline comments and docstrings explaining your code.
- Ensure the code runs without errors.
- Work collaboratively with your group.
- Submit before the deadline on MyCourses.
- Prepare to present your solution in class.