GENG0033 COURSEWORK - CSFY 22-23

LABORATORY WORK 2 – PYTHON 3.0 (GUI)

1. THIS COURSEWORK WAS DEVELOPED TO ACHIEVE MODULE LEARNING OUTCOME(S):

CO2: Demonstrate knowledge and understanding of digital literacies and use of common digital tools

CO3: Able to apply problem-solving techniques to familiar and unfamiliar problems

2. GENERAL NOTES:

- This is an individual basis of laboratory work with logbook writeup as detailed in the laboratory handbook.
- This laboratory work contributes 12% to the final mark weighting of the final coursework mark.
- Your code should be clean and easy to read and understand. You should minimize unnecessary redundancies.
- You can show your distinctive skills by experimenting with new and different things!
- **Submit ONLY** to **the Blackboard** submission link unless you do not have a SOTON account.

3. TASKS:

You want to develop a **Simple Email System**, like Outlook, Gmail, etc, where users can send/receive emails to/from other users. The main focus of this coursework is the GUI and the functionality; forget about the security and complexity.

The email system will have the following features:

a. Graphical User Interfaces (GUI)
Python provides diverse options for developing GUIs, such as Tkinter.

b. Email Database
You should create a simple text file that will become the database for the email. The file contains all the emails; an email consists of **from**, **to**, **title**, and **content**. You may decide how to format the text file. Below is an example text file format, with \n (new line or enter) as the separators:

```
rose@mymail.com
violet@mymail.com
Payment Invoice
```

```
Hi Violet, attached is the invoice for the cleaning service today
Regards, Violet

prime@mymail.com
phil@mymail.com
Lab Work Solution
Hi Phil, the lab work seems to be very difficult, I can't do it.
Regards, Prime
```

In the example above, each email is separated with \n\n (double newlines). For each email, the first line is the **form**, followed by **to**, **title**, and the rest of it (until \n\n) is the **content**. Of course, this format is not perfect (but it's ok for now); inputting a message with double Enter (newlines) will cause a problem when the program is trying to read the messages. We just assume that this will not happen. Again, you may choose the format that you want. Refer to the following references that may help you:

- Reading and writing files: https://www.geeksforgeeks.org/reading-writing-text-files-python/
- String .split(): https://www.w3schools.com/python/ref_string_split.asp

c. Register (and the User Database)

There will be a GUI where the user can register to the email system, by entering the **email address** and the **password**. Similar to the email database, you should create a simple text file that simulates a database for the user's data. There should be no duplication of the email address. Example of the user's text file (it uses comma as the separator for address and password, and \n for different accounts):

```
rose@mymail.com,abc123
prime@mymail.com,jhC23#
```

d. Login

There will be a GUI where the user can enter their **email address** and **password**. The data will then be validated using the user database.

e. Sending Email

After login, there will be a GUI where the user can enter the **from**, **to**, **title**, **content**, and also **Send** buttons. Sending an email means registering the email data into the email database mentioned above. When the email is sent successfully (an entry registered in the text file), there will be a notice like "Email sent successfully".

f. Checking Mail Box

After login, there will be a GUI where the user can check their mailbox. The GUI will show all the emails (including the **from**, **title**, and **content**) that have the **to** field equal to the current login information.

4. REPORT

To complete this coursework, you need to evaluate your own program. You will need to run and test your program and give a critical analysis of the program. These are the important points that you need to investigate while analysing your program:

I. **Introduction**
   A brief introduction on how the system works.

II. **Quality of work**
   The functionalities of the program work as per specifications by checking screens and controls like menus, buttons, icons, etc.

III. **Program Design**
   The program design uses appropriate structures. The overall program design is appropriate.

IV. **User Interface Design**
   User interaction is as specified and is natural to the user. You do not need help while you are using the system.

V. **Solution**
   The program is a complete solution that runs without errors. It meets all the specifications and works for all test data.

VI. **Suggestions for improvement**
   Your suggestion to your partner if there is anything that she/he could do to improve the program.

VII. **Conclusion**
   After you have run and tested the program, what is your overall experience of using the program or as an evaluator of the program? Did you learn something from the task, or would you improve your own program by adopting a few of the functions from your partner?

Allow 1 page for each section. And your report should not be more than 10 pages.

5. SUBMISSION:

You need to submit these to the blackboard.

i. Report
ii. Phyton File
iii. Presentation Video Link

You will need to record your demonstration of the program for a maximum of 7 minutes duration.

6. MARKING SCHEME

Criteria	Marks
Meeting system requirements GUI Design User Interaction System requirements	 10 10 20
The application Code complexity, sufficient comments, meaningful variables and functions	20
The initiative, innovation, creativity and effort, problem-solving	10
Presentation (Recording)	10
Self Evaluation Report The report is written following the given guidelines/format.	20
Total marks	100

7. MARKING SCHEME

You are expected to show how you developed your understanding and to demonstrate scientific curiosity and reflection in ensuring you fully understand your submission.

<40%	You have not succeeded in coding with Python programming
40-49%	You have got some of the Python code workings
50 -59%	You appear to have got your Python codes working but your code standard (comments, meaningful variables and functions) and report is too brief and unconvincing
60-69%	You have got the program working and some sensible, well-written notes in the report showing a good understanding of the given tasks.
70-80%	Your report demonstrates an excellent understanding of, and reflection on, the given tasks, and you have, where appropriate, completed beyond those required tasks.

8. PROFESSIONAL CONDUCT:

You must refer and read to the University Regulations 2021-22: Regulations governing Academic Integrity for the details of professional conduct.

9. LATE SUBMISSION POLICY:

- The University has a uniform policy for the late submission penalty for a piece of assessed work worth 10% or more of the final module mark.
- Work submitted up to 5 days after the deadline should be marked as usual, including moderation or second marking, and feedback prepared and given to the student. The final agreed mark is then reduced by the factors in the following table.

University Working Days late	Mark
1	(Final agreed mark) *0.9
2	(Final agreed mark) *0.8
3	(Final agreed mark) *0.7
4	(Final agreed mark) *0.6
5	(Final agreed mark) *0.5
More than 5	Zero

Work submitted late and accompanied by a special considerations form should be marked as usual and the penalties in the table above applied.

Grading Criteria	Wtg	Exceptional 4	Acceptable 3	Amateur 2	Unsatisfactory 1	0	Mark (Wtg x Score)
GUI Design	**2.5**	Excellent selection of graphic elements that enhance the understanding, concepts or ideas. Great choice of layout and design.	Appropriate and thematic selection of graphic elements that enhance the understanding of the concepts or ideas. Variation in layout or design.	Graphics or maps are included but have little variations in layout or design. Background colours or template design is acceptable.	Few graphics are included. No variation in layout or design. Background colours or template design is distracting.	No graphics are included. GUI design and theme are not consistent.	**/10**
User Interaction	**2.5**	User interaction is as specified and is natural to the user.	User interaction generally meets the specifications and is acceptable to the user	User interaction minimally meets the specifications but does not increase the usability	User interaction is incomplete and does not meet specifications.	Impossible for any user to interaction.	**/10**
System Requirement	**5.0**	The program works exceptionally and beyond all of the specifications listed	The program works and produces the correct results and displays them correctly. It also meets all of the other specifications	The program produces correct results but does not display them correctly and meets almost all of the specification	The program is producing incorrect results with many errors and meets only few of the requirements	The program does not meet any of the requirement	**/20**
Application	**5.0**	The code is exceptionally well organized and very easy to follow. Exceptional comments with meaningful variables and functions	The code is fairly easy to read. Good comments with some meaningful variables and functions	The code is readable only by someone who knows what it is supposed to be doing. Fair comments with few meaningful variables and functions	The code is poorly organized and very difficult to read. No comments with no meaningful variables and functions	The code is not organized	**/20**

The initiative, innovation, creativity and effort, problem-solving	**2.5**	The code is extremely efficient without sacrificing readability and understanding. Shows great initiative, innovation and very creative effort in developing the program	The code is fairly efficient without sacrificing readability and understanding. Shows initiative, innovation and creativity and good effort in developing the program	The code is brute force and unnecessarily long. Shows fair initiative, innovation and fair creativity and less effort in developing the program	The code is huge and appears to be patched together. Shows no initiative, innovation and no creativity or effort in developing the program	The program is incomplete	**/10**
Demonstration Video	**2.5**	Includes more than required in the presentation Presents ideas in exceptional order Organizes time very well. Audience managed to really understand the program through the demo and liked the program	Includes everything required in the presentation Presents ideas in order that makes sense Organizes time well. No part of the presentation is rushed, too short or too long. Audience managed to understand the program through the demo and liked the program	Includes almost everything required in the presentation Tries to present ideas in order but it doesn't always make sense Presents for the right length of time, but some parts may be too long or too short Audience managed to fairly understand the program through the demo	Does not include everything required in the presentation Presents ideas in an order that does not make sense Does not plan the timing of the presentation well. It is too short or too long Audience not managed to understand the program through the demo	No video attached	**/10**
Report **(Introduction)**	**1.25**	Very well written. Manage to make the reader fully understand the objective of the program and be interested to read more.	Well written. Uses proper terminology to introduce the program. Manage to make readers understand the objective of the program.	Well written in basic form. Uses minimal terminology to introduce the program. Manage to make the reader partially understand the objective of the program.	Written in basic form. Does not use appropriate terminology to introduce the program. The reader was not able to understand the objective of the program and not interested to read more.	No introduction was written.	**/5**
Report	**1.25**	The analysis is written in an excellent manner.	The analysis is well written. It shows that	The analysis is written. It shows that some of the	The analysis is poorly written. It shows that	No report on this part	**/5**

(Quality of work, Program Design and GUI, solution)		It shows that every component has been considered and analysed.	most of the components have been considered and analysed.	components have not been considered and analysed.	most of the components are not been considered and analysed.		
Report **(Suggestions for improvement)**	**1.25**	Written in very detail with the support of evidence such as finding from the program test.	Well written with the support of evidence such as finding from the program test.	Written without the support of evidence such as finding from the program test.	The suggestion is unclear.	No suggestion.	**/5**
Report **(Conclusion)**	**1.25**	A detailed discussion of all aspects of the test, results and suggestions.	Discussion of most aspects of the test, results and suggestions.	Discussion of some aspects of the test, results and suggestions but somehow lacking details.	Discussion of a few aspects of the test, results and suggestions but somehow missing details.	No conclusion	**/5**
						TOTAL	**/100**