GENG0033 COURSEWORK (PROGRAMMING AND COURSEWORK) - CSFY 22-23

LABORATORY WORK 1 – PYTHON 3.0 (FUNCTIONS)

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 redundancy.
- 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 are required to write a complete program using Python 3.0. Write an inventory program to be used for any store such as a pharmacy, groceries store, restaurant or any type of business that may have an inventory system. For example, the program should be able to allow users to choose which task to be done from a selection of tasks (function):

- To add a new item with quantity and price
- To remove or delete an item
- To view the item (this is to check on the quantity left for stock count purposes or to check on the expiry date)
- To search for item(s)

These are suggestions but are not limited to the listed functions above. You may add accordingly. As it is an inventory program, try using a text/ binary file, CSV file or SQLite file to store the data instead of using a list. You may use libraries such as pandas (Example reference: https://realpython.com/pandas-read-write-files/#write-a-csv-file) for data management.

To develop a complete program, ensure to include the following requirements in your program:

- Selection statement
- Repetitive statement (suggestion: to allow the user to enter details and to decide whether to continue with the program or to exit the program)
- Functions where applicable.

This system may not limit to the above requirements only. You may cover many options based on your idea. Remember, be creative!

4. REPORT

To support your program, you need to write 8-10 pages of a full report to explain the details of your program. Screenshots should be included in the report and counted against the page limit. The report needs to include the following:

I. Introduction
II. Detail explanation of each function
III. Screenshot of coding and output
IV. Future work (Suggestion for improvement)
V. Conclusion

5. DEMONSTRATION VIDEO

To support your program, record a demonstration video of your program for a maximum of 10 minutes. Run the program, key in input from the user and show the output. Imagine that you are promoting this system to a potential buyer, that’s how you should explain your program. You may use Zoom or Teams to record your demonstration video.

6. MARKING SCHEME

Criteria	Marks
Meeting system requirements	40
Documentation The report is written following the given guidelines/ format.	20
The application A code standard, sufficient comments, meaningful variables, functions	20
The initiative, innovation, creativity and effort, problem-solving	10
Demonstration video (10 minutes)	10
Total marks	100

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.

7. PROFESSIONAL CONDUCT:

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

8. 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.

Jniversity Working Days late	Mark
1	(Final agreed mark) *0.9
2	(Final agreed mark) *0.9
3	(Final agreed mark) *0.9
4	(Final agreed mark) *0.9
5	(Final agreed mark) *0.9
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)
Specification	**10.0**	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 a few of the requirements	The program does not meet any of the requirement	**/40**
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**
Documentation	**5.0**	The documentation is well written and clearly explains what the code is accomplishing and how with adequate screenshot	The documentation consists of embedded comments and some simple header documentation that is somewhat useful in understanding the code with an acceptable amount of screenshot	The documentation is simply comments embedded in the code with some simple header comments separating routines with a fair amount of screenshot	The documentation is simply comments embedded in the code and does not help the reader understand the code with few screenshots without explanation	No documentation	**/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. The 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. The 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**
						Total Marks	**/100**