BSc (Hons) Computing

Module Code: QH0305 Module Title: Problem Solving

Assessment Sheet 4

Instructions:

This is one of the eight assessment tasks, which will contribute to the overall mark. You will need to complete the tasks as outlined below and then document them in a Word document. As a minimum, you should provide screenshots of the following:

- Your code
- The output that your code generates

In instances where your code could generate different outputs (depending on what values it is given), you should provide multiple screenshots of the console screen, showing the different outputs to demonstrate that the code is working correctly.

This assessment will focus on all the concepts you have learnt up until now.

You must attempt all tasks on this sheet to achieve a higher grade. For example, if you want to gain Grade A, you must complete all other grades first and add them to your portfolio with screenshots.

A zip folder with all Grade codes must be attached inside of the portfolio (MS Word file).

Task 4: Smart plant care system

In this task, you are required to create a program that simulates a smart plant care system. The program should provide a menu to the user with the following options:

a. Add a new plant to the system
b. Update plant information
c. Search for a plant
y. Exit

Please note that the full program will have all these options. However, certain grades will contain only a part of this menu. The program should utilise loops and conditional statements to ensure the proper flow of actions.

Note: Do **not** use functions in this assessment as those are covered in Assessment Sheet 7.

To achieve a D grade

Only the menu functionality “a. Add a new plant to the system” is needed to achieve grade D.

a. Add a new plant to the system:

- Prompt the user to enter the plant's name, species, watering schedule, and sunlight requirements.
- Generate a unique ID for the new plant.
- Store the plant details (ID, name, species, watering schedule, and sunlight requirements) using individual variables. For example, you can define variables such as plantID for the unique plant ID, plantName for the plant's name, plantSpecies for the species of the plant, wateringSchedule for the watering schedule for the plant (e.g., daily, weekly), and sunlightRequirements for the sunlight requirements for the plant (e.g., full sun, partial shade).

- You need to provide screenshots of your code and the different outcomes it can give based on user inputs.

- Your Word document should have appropriate headings to ensure that this task can easily be identified alongside the rest of your work.

To achieve a C grade

Complete all previous steps, then modify the above program so it behaves as follows:

b. Update plant information:

- Prompt the user to enter the plant's ID to update.
- Check if the plant ID exists.
- If it exists, allow the user to update the plant's name, species, watering schedule, or sunlight requirements.
- If it does not exist, display an appropriate error message.

c. Search for a plant:

- Prompt the user to enter the name of the plant they want to search for.
- If the plant is found, display its details.
- If not found, display a message indicating that the plant is not in the system.

Complete all previous steps, then modify the above program so it behaves as follows:

To achieve a B grade

Complete all previous steps, then modify the above program so it now behaves as follows:

- Implement error handling for invalid inputs. The program should inform the user if they entered an invalid menu choice. For example, if the user entered 'f', then the program should say "Sorry, your input is invalid. Please enter either a, b or c."
- Accept both upper-case and lower-case letters for menu options.
- In all instances, the program should behave in a user-friendly manner. All outputs should be accompanied by a text and all scanf() function calls should be accompanied by a printf() letting the user know what they must do.
- Currently, the program will end once the user has performed their chosen action. If they wanted to select another action, they would have to run the program again. You will need to modify the above program so that once the user has completed their chosen action, instead of ending, the program will display the menu again and thus allows the user to select another option. This will require a **while loop**.
- You should also add the following option to the menu: "y – Exit program" If the user enters ‘y’ (either upper-case or lower-case), the program should stop looping and it should end.
- As always, with chars, the program may behave erratically: this should be fixed to ensure proper validation and error handling for all menu options.

To achieve an A grade

Complete all previous steps, then:

- All tasks must be accompanied by written descriptions or annotations. These must demonstrate satisfactory understanding of what the code does.
- To achieve higher grades, there is always a requirement of some independent research. Implement pointers in the program and explain their suitability. This should include how pointers improve the program's efficiency and flexibility in handling dynamic data structures.

Assignment Preparation Guidelines

- All components of the assignment report must be Word-processed **(handwritten text or hand drawn diagrams are not acceptable)**, font size must be within the range of 11 point to 14 point including the headings, body text and any texts within diagrams.
- Standard and commonly used fonts such as Times New Roman, Arial or Calibri should be used.
- All figures, graphs and tables must be numbered and labelled with short explanations.
- Material from external sources must be properly acknowledged and cited within the text using the Harvard referencing system.
- All components of the assignment (text, diagrams. code etc.) must be submitted in one-Word file.
- The report should be logically structured; the core of the report may start by defining the problem / requirements, followed by the proposed solution including a detailed discussion, analysis and evaluation, leading to the implementation and testing stage, finally a conclusion and a personal reflection on learning.
- Screenshots without description / discussion are not suitable as they do not express your understanding or support your work adequately.

Submission instructions

- This is a portfolio assignment with eight tasks in total. Each task will be completed and saved in the portfolio. Once the portfolio is completed, it should be submitted on Turnitin. The submission link to Turnitin can be found under the “Assessment Tab” in your module section in the SOL VLE.

- Please note that file size limitations might apply. Your report must be under 250MB.

- The source code for each task should be **zipped** and **attached** to your Word document report submission in the appendix.

- The Assignment Brief can be found under the “Assessment Tab” in your module section in the SOL VLE.

- **Refer to the Assignment Brief** to find the links to Late Submissions, Extenuating Circumstances, Academic Misconduct, Ethics Policy, Grade marking and Guidance for online submission through Solent Online Learning (SOL).