COMP0015 Coursework Brief

Description

This document explains the arrangements for the coursework. You will create a **Chore Chart**. The purpose of the application is to help housemates (people sharing a house) to keep a record of what needs doing every week and who is doing it. There is a leaderboard that shows who has earned the most points for household tasks so far[1]. The project runs through the remainder of the term and counts for 65% of the marks available for the module. The lab sessions can, and should, be used by you to work on your project and discuss the work with teaching assistants.

You will deliver the project in three stages, broadly speaking these will contain functionality to:

1. Register a household and the household tasks.
2. Enter the tasks completed.
3. Use a text file to store and retrieve the information and display a leaderboard.

The coursework is phased and modular so that you:

- do not have a single, large deliverable at the end of term,
- can get helpful feedback during development and,
- will gain experience of developing a non-trivial application composed of a number of python classes.

Deliverables

1.1 Deliverable 1: Main Menu and Create a Household – 20%

You must deliver an application that can be used to enter information about a household. When your application starts, the user should see a menu with the following options:

```
Welcome to Chore Chart

     About                 (A)
     Create Household      (C)
     View Household        (V)
     Log Chores Done       (L)
     Show Leaderboard      (S)
     Quit                  (Q)

     Please choose an option and press <Enter>: _
```

The rules governing the main menu:

1.	Choosing option 'A' will display an explanation for the application.

[1] The idea is inspired by a UCL student, Iustin Targovet and Ariel Procaccia in his work on Social Economics.

2.	At the moment, choosing 'L' or 'S' for this deliverable will take the user back to the main menu. Choosing 'Q' will end the application
3.	Choosing option 'C' will allow the user to create a new household using the dialogue below. The user will enter a blank line when they have finished adding participants or chores.
4.	Choosing option 'V' will allow the user to view the household information.
5.	The user should be able to view the last household entered before quitting the program.

The dialogue for option 'C':

Choosing option 'C' will allow the user to create a new household. The dialogue is shown in Figure 1, information entered by the user is underlined, you should not reproduce the underline in your application:

```
Enter the household name: Penguins
Enter participants' names:

      Enter the name of participant 1: Xiang
      Enter the name of participant 2: Asim
      Enter the name of participant 3: Bogdan
      Enter the name of participant 4: _

Enter chores:
      Chore 1: Hoover the hallway
            Times per week : 1
      Chore 2: Wash up
            Times per week : 5
      Chore 3: Empty the bin
            Times per week : 2
      Chore 4: _

Press <Enter> to return to the main menu: _
```

Figure 1 - Dialogue for option 'C'

The dialogue for option 'V':

Choosing option 'V' will allow the user to view the household information. The dialogue is shown in Figure 2, information entered by the user is underlined, you should not reproduce the underline in your application:

```
View Household:

Penguins

Participants:
      1. Xiang
      2. Asim
      3. Bogdan

Weekly Chores:
      1. Hoover the hallway (5)
      2: Wash up (5)
      3: Empty the bin (2)

Press <Enter> to return to the main menu: _
```

Figure 2 - Option 'V'

Coding style is important, please see the section on Coding Style, below.

It is essential to test and validate your software carefully. You can do this by entering invalid information and seeing how your program behaves. For example: does your program accept the value '-1' for the number of times that a chore must be done every week?

Note: you are expected to show that you can code competently using the programming fundamentals covered so far in the course including (but not limited to): conditions, loops, functions and lists.

Due to the iterative nature of software development it is highly likely that your software will alter by the end of the term. This is normal and completely acceptable. However, any changes you make will not affect the mark given for this deliverable.

1.2 Deliverable 2: Using classes and logging chores done – 40%

In this iteration of your application, you will use the following classes:

1. Household class – to manage all the information about a Household.
2. ParticipantsList class – to manage all the information about the participants.
3. ChoresList class – to manage all the information about the list of chores.

These classes are provided for you on the Codio platform. Some are partially completed and you are required to complete the classes.

You must implement the menu options: 'L' for logging the chores done by and 'S' for showing the Leaderboard.

The rules governing options 'V', 'L' and 'S':

1.	Choosing option 'L' will allow the user of the application to log the chores that a household member has completed. The number that is entered will be added to the previous total.
2.	Choosing option 'S' will allow the user of the application to show the leaderboard for chores that all the household members have completed.

The dialogue for option 'L':

Choosing option 'L' will allow the user to log chores for an individual. The dialogue is shown in Figure 3, information entered by the user is underlined, **you should not reproduce the underline in your application**:

```
Log Chores

Participants:
      1: Xiang
      2: Asim
      3: Bogdan

Enter the participant number: 3

You are logging Bogdan's chores.

Chores:
      1: Hoover the hallway
      2: Wash up
      3: Empty the bin

Enter the chore number: 3

Bogdan has done 'Empty the bin' 3 times.
How many more times has Bogdan done 'Empty the bin`: 7

Bogdan has done 'Empty the bin' 10 times.

Press <Enter> to return to the main menu: _
```

Figure 3 - Dialogue for option 'L'

The dialogue for option 'S':

Choosing option 'S' will allow the user to view the leaderboard. The dialogue is shown in Figure 4. Note: the order of participants is not important. Information entered by the user is underlined, **you should not reproduce the underline in your application**:

```
Leaderboard:

Bogdan:
      Hoover the hallway (22)
      Wash up            (0)
      Empty the bin      (2)

Xiang:
      Hoover the hallway (2)
      Wash up            (20)
      Empty the bin      (15)

Asim:
      Hoover the hallway (0)
      Wash up            (54)
      Empty the bin      (0)

Press <Enter> to return to the main menu: _
```

Figure 4 - Dialogue for option 'S'

Submission requirements

For all three deliverables you will use the Codio system to develop and submit your code. Team members must have exactly the same code files in their project folders so that either student's Codio account can be used for marking. The deliverables are:

- A short design report (1 page maximum) and,
- Your source code.

The design report is not marked, its purpose is to help guide the marker. It should be submitted on moodle and should contain:

- Brief instructions on how to run your program,
- The Python style guide that you used,
- A short description of how you tested your app,
- A contribution mark for each team member; you have 20 points to share between you. This mark will only be used by the course leader if there is a significant difference in the contribution of either team member.

Assessment

Your project work will be marked using the rubric made available to you on moodle. Team will receive the same mark unless the lecturer has been notified of issues and determines otherwise.

Marks for your project work will be awarded for the capabilities (i.e. functional requirements) your system achieves, evidence of the design process you followed and the quality of the code. You should show that you have thought about the structure of your application, considering where the logic for the various elements of the application should reside. You should also demonstrate that you have thought about how the functionality should be broken down into various components of the application.

Coding Style

Coding style is important: well-formatted code, self-explanatory variable names and comments enhance the readability of programs. Whilst there are no universally accepted coding style rules, developers will generally agree on which guidelines to use at the start of a project. You should choose a set of guidelines, you may use either of these:

Short – Python Style Guide, Simplified version for beginner programmers by John Magee

The real thing - PEP8 – Style Guide for Python Code

Additional Challenges

The most effective way to improve your grade is not to take on additional challenges but to ask a member of staff for feedback in lab sessions. This is because staff can provide information about how to improve your application design.

Additional marks may also be gained by taking on extra challenges but you should only do this if you have satisfied all requirements for the deliverable you are working on. You can think of your own challenges or you might like to consider:

- Automating testing with unittest, this is arguably the most important additional thing you can learn.
- Allowing the user to edit or delete the information for a household or extending the functionality of the application in some other way.

- Using Github (see Appendix 1)
- Using a database.

Appendix 1

Using Github

!!! Warning: Do not use Github unless you have significant time to spare.

Github is a professional tool used by software engineers to manage or control versions of code or other material and to share and collaborate with others. You might like to use your project development as a way of getting to know how to use it. Be warned, it can be complicated.

First Steps with Github: If you don't already have a GitHub account then you can create one using the free Student Developer Pack option: https://education.github.com/pack. Using an education account will allow you to create repositories that are private. This is important as you won't want others to be able to see your code. GitHub Guides explain how to create a repository: https://guides.github.com/activities/hello-world/. Next you should add your team member as a collaborator for the project so that they can use the repository too. Go to the Settings for the newly created repository, click on Collaborators and add the GitHub usernames of the others in your team. https://help.github.com/articles/inviting-collaborators-to-a-personal-repository/.

Create and merge a branch of the code: Using the 'Hello World' GitHub Guide, all members should create a branch of your group repository, make a simple change, commit it, create a pull request and finally merge the change. See here: https://guides.github.com/activities/hello-world/.

Github and Codio: There are some instructions for associating your Codio account with your Github account on this page, navigate to the section entitled: '**I do not yet have a remote repo**'. Once you have done that, you can open a terminal and use git commands to check out the project, create branches, commit changes and so on. Use the Github flow described here: https://guides.github.com/introduction/flow/. You will use the Codio terminal command line and type git commands to manage your git repository. One of the best guides I have found is this one: https://services.github.com/on-demand/github-cli/, although the examples featured are web pages not code.