## Coursework general instructions:

### Code commenting and documentation

Your code must be documented appropriately using docstring comments. Document as you go, not all at the end. One strategy is to write a brief comment about the "one thing" that the function is supposed to do when you declare it, then refer to that comment while implementing it: this helps maintain focus when implementing the function and helps stop yourself from making it do more than the 'one' thing it is supposed to do. Each class and each method should have at least one docstring which can be brief.

For this project you will need to create a **readme.md** text file in your ~/project folder that explains each part of the project. Specifications are below:

- **Part 1**: readme.md file should include a description of your code design, classes, methods, and other key details. Your readme.md file should be at least several paragraphs in length and should explain what your project is, what each of the files you wrote for the project contains and does, and if you debated certain design choices, explain why you made them. Ensure you allocate sufficient time and energy to writing a **readme.md** that documents your code thoroughly.

- **Part 2**: Documentation for part 2 is in Jupyter notebook using markdown cells. Any additional libraries, external sources needed to run the code correctly must be explained in the **readme.md** file.

If you are unfamiliar with Markdown syntax, you might find GitHub's Basic Writing and Formatting Syntax helpful:

https://docs.github.com/en/free-pro-team@latest/github/writing-on-github/basic-writing-and-formatting-syntax

## Version Control

You must use version control to keep track of your progress during implementation using Git, specifically GitHub. You should perform regular commits as you implement features and fix errors. Your commit comments should reflect the context of the changes that were made in each commit—a fellow developer can always diff the contents to see exactly what changed but that does not provide the context. You should try to ensure each commit comment contains a short subject line. Ensure that the repository is private, i.e., cannot be seen by people other than yourself. Failure to do that will be considered plagiarism. If version control is not submitted or made public, 10 pts will be deducted.

## Submitting your coursework

Your submission will comprise your entire version control repository for the three parts. Your repository must contain all source files required to run your program.
Your submission must include your code for each part: part 1 and part 2.

- **Part 1**: Includes different Python scripts to run your code.
- **Part 2**: Jupyter Notebook file that includes code and explanation in the markdown for each step. The collected data saved in Step 1 must be submitted too.
- **GitHub markdown (readme.md file)**: to explain any required instructions for running your code (please refer to "code documentation and commenting section").
- If you have used any additional code for part 2, beyond standard Python packages, then you will need to include an additional subdirectory containing additional code required to run your scripts. The authorship of any such code should be clearly stated in the GitHub markdown (**readme.md**).
    - Note: A link of GitHub repository must be included in your readme.md file.

## Grading

Your grade will be assessed on both correctness and style.

- **Correctness**: How well the program *works according to specifications. How good is the quality of your analysis*.
- **Style**: The quality of the code and documentation.

Your code should be well-written and well-commented. You are encouraged to format your code per Python style guidelines (https://www.python.org/dev/peps/pep-0008/). It should be clear enough for another programmer, such as the course staff, to understand and modify it if needed. Quality code is expected to meet our style requirements:

- Every function written is commented, in your own words, using a docstring format that describes its behaviour, parameters, returns, and highlights any special cases.
- There is a comment at the top of each code file you write with your name, section, and a brief description of what that program does.

## Part 1: Software Development (66.7%)

The part will assess whether you can program competently in Python, using object-oriented techniques (ILO 1). You will have to design, implement, test, and debug a text-based fish hatchery simulation. You must use an object-oriented approach with appropriate relationships between classes.

This simulation must be text-based and will not have a graphical user interface; hence it is not allowed to use turtle, pygame, tkinter, matplotlib or any other API or external library for implementation. Only python standard libraries can be used in Part 1 https://docs.python.org/3/library/

### Background

A fish hatchery sells several types of fish species that customers can purchase by the pond (this is the unit for fish when buying, selling, restocking). The hatchery needs to have employees to maintain and sell the fish. It also needs to purchase supplies that differ for each species. Each quarter the hatchery manager must decide whether to add/remove aquaculture technicians, species to raise and sell, pay expenses, and choose which vendor to restock supplies. The goal of the hatchery is to make a profit and avoid going bankrupt.

### Task overview

In this part, you are required to write a text-based (no GUI is required) Python program using object-oriented programming to simulate the quarterly actions of the shop owner. Your program should do the following:

A. The hatchery sells the following species requiring the specified resources.

| Fish Type | Fertilizers (ml) | Feed (kg) | Salt (kg) | Maintenance Time (in days) |
|---|---|---|---|---|
| **Clef Fins** | 100.0 | 12 | 2 | 2.0 |
| **Timpani Snapper** | 50.0 | 9 | 2 | 1.0 |
| **Andalusian Brim** | 90.0 | 6 | 2 | 0.5 |
| **Plagal Cod** | 100.0 | 10 | 2 | 2.0 |
| **Fugue Flounder** | 200.0 | 12 | 2 | 2.5 |
| **Modal Bass** | 300.0 | 12 | 6 | 3.0 |

The hatchery has two warehouses, the main and auxiliary. The following table shows the maximum quantity that the hatchery's warehouses can store in any given quarter. The warehouse costs are the same for both main and auxiliary warehouses. The table shows the losses that occur each quarter due to depreciation (e.g. feed going bad, we assume salt does not go bad). When determining the quantity to depreciate, be sure to take the ceiling (i.e. round up to nearest integer). The table also shows the warehouse costs per quarter. For example, if the warehouse has 300 litres that did not get used during the quarter, only 180 litres will be available for the next quarter and the hatchery will have incurred warehouse costs of £30.

| Supply | Main Capacity | Aux Capacity | Depreciation | Warehouse Costs |
|---|---|---|---|---|
| **Fertiliser** | 20 litres | 10 litres | 0.4/quarter | £0.10 / litre |
| **Feed** | 400 kg | 200 kg | 0.1/quarter | £0.001 / g |
| **Salt** | 200 kg | 100 kg | 0.0/quarter | £0.001 / g |

The hatchery pays a fixed quarterly cost of £1500 and keeps a quarterly amount of cash. The hatchery starts with a cash balance of £10,000.

B. Regardless of how much demand there is, each technician gets paid per quarter (~ twelve weeks) and provides nine weeks of labour per quarter. Each technician has a name and works at a fixed rate of 500 pounds per week. You can only add/remove technicians each quarter (i.e. even if there is no demand, you must pay technicians for the entire quarter). Based on statutes, the maximum number of technicians that can be added is 5 and there must be at least one hired in the quarter. Assume the manager requires no salary and cannot sell any fish.

C. There two vendors that sell the supplies as follows. Assume the vendor can deliver the supplies immediately and that there is no limit to the amount than can be purchased.

| Supplier Name | Fertilizers | Feed | Salt |
|---|---|---|---|
| **Slippery Lakes** | £0.30/litre | £0.10/kg | £0.05/kg |
| **Scaly Wholesaler** | £0.20/litre | £0.40/kg | £0.25/kg |

D. The customer demand by species per quarter is shown in the following table.

| Fish Species | Demand | Price |
|---|---|---|
| **Clef Fins** | 25 | £250 |
| **Timpani Snapper** | 10 | £350 |
| **Andalusian Brim** | 15 | £250 |
| **Plagal Cod** | 20 | £400 |
| **Fugue Flounder** | 30 | £550 |
| **Modal Bass** | 50 | £500 |

E. At the beginning of the program, prompt the user to enter the number of quarters to run the simulation. Default will be to run the simulation for two years. Assume the hatchery's warehouse is full when the simulation starts.
F. Allow the manager to choose how many technicians to add/remove and the number of fish to sell for each species (these must be positive integers). When removing a technician, you are free to decide which technician is removed but be sure to indicate which one was removed.
G. For each quarter:
    a. Prompt the manager to add/remove technicians at beginning of the quarter.
    b. Prompt the manager for how much of each fish type to sell, ensuring that it does not exceed the demand, supplies, and labour constraints.
    c. Given these decisions, assume the quarter runs accordingly. Based on how much was sold, update the hatchery's cash.
    d. Pay the technicians from cash.
    e. Pay supply costs from cash.
    f. Display the status of the hatchery to include the number of technicians, the technician's names (any specialities), and number of supplies in each warehouse.
    g. Apply the depreciation to the warehouses.
    h. Prompt the manager to select a vendor to purchase supplies from.
    i. Purchase supplies from the supplier so that the pantry is fully replenished.
    j. If there is not enough cash on hand to pay expenses, then the hatchery goes bankrupt and the simulation ends.

The simulation stops after the quarters specified, or the hatchery goes bankrupt.

## Task specifications

- **Create your classes (30% of Part 1)**
  Your program should have at least three classes, in their own file, to include:
    - **Hatchery class** contains attributes and associated functions about the supplies, cash status, and number of technicians.
- **Run your code (50% of Part 1)**
  Create the main script (main.py) to run the classes you implemented. The main script must not define any classes. The classes should be imported into the main script. All constraints must be met. The simulation ends when the maximum number of quarters or the hatchery goes bankrupt.
- **Extend your program (20% of Part 1)**
  Make sure the basic requirements are implemented correctly before extending your program. Modify your program so that a technician can be specialised in maintaining/selling a certain type of fish. They can still sell other fish types, but they can maintain/sell the specialised fish in 2/3 the time. When hiring a technician, prompt for whether the technician has a specialty, and if so, prompt for the fish type they specialise in. Update so that the fish demand and technician specialty is considered when meeting customer demand. Demand for a fish type is first met by specialised technicians followed by regular technicians.

- **Errors and exceptions handling (included in the grading for each step)**

Since you are taking user input, your code will need to handle potential errors. Most errors occur in Null values and non-integer inputs. Make sure you handle these errors. Your program should continue to run when given invalid input. Examples of errors to be handled:

- No input
- Non-integer or positive input
- A technician with same name already exists

## Example Run:

Below is an example of how to interact and what you expect from the basic version of your text-based simulation, be sure to also add the extension. You may use different messages and output, just be sure that it is user friendly and meets the requirements.

```
$ python3 main.py >>> Please enter number of quarters: 8
================================
====== SIMULATING quarter 1 ======
================================
To add enter positive, to remove enter negative, no change enter 0.
>>> Enter number of technicians: 2
>>> Enter technician name: James Hired
James, weekly rate=500 in quarter 1
>>> Enter technician name: Richard
Hired Richard, weekly rate=500 in quarter 1
Fish Clef Fins, demand 25, sell 25: 25
Fish Timpani Snapper, demand 10, sell 10: 10
Fish Andalusian Brim, demand 15, sell 15: 15
Fish Plagal Cod, demand 20, sell 20: 20
Insufficient labour: required 8.00 weeks, available
4.50 Insufficient ingredients:    fertiliser need 2.0
storage 25.65    feed need 200.0 storage 120.0    salt
need 40.0 storage 200.0 Fish Plagal Cod, demand 20, sell
20: 0
Fish Fugue Flounder, demand 30, sell 30: 0
Fish Modal Bass, demand 50, sell 50: 0
Paid James, weekly rate=500 amount 6000
Paid Richard, weekly rate=500 amount 6000
Paid rent/utilities 1500
Warehouse Main: Fertiliser cost 1.57
Warehouse Main: Feed cost 0.00
Warehouse Main: Salt cost 100.00
Warehouse Auxilliary: Fertiliser cost 1.00
Warehouse Auxilliary: Feed cost 120.00
Warehouse Auxilliary: Salt cost 100.00
List of Vendors
1. Slippery Lakes
2. Scaly Wholesaler
>>> Enter number of vendor to purchase from: 1
Hatchery Name: Eastaboga, Cash: 9618.63
    Warehouse Main
        Fertiliser, 20.00 (capacity=20)
        Feed, 400.00 (capacity=400)
        Salt, 200.00 (capacity=200)
    Warehouse Auxilliary
        Fertiliser, 10.00 (capacity=10)
        Feed, 200.00 (capacity=200)
```

```
        Salt, 100.00 (capacity=100)
    Technicians
        Technician James, weekly rate=500
        Technician Richard, weekly rate=500
END OF QUARTER 1
================================
====== SIMULATING quarter 2 ======
================================
To add enter positive, to remove enter negative, no change enter 0.
>>> Enter number of technicians: -1
Let go Richard, weekly rate=500 in quarter 2
Fish Clef Fins, demand 25, sell 25: 25
Insufficient labour: required 10.00 weeks, available 9.00
Fish Clef Fins, demand 25, sell 25: 0
Fish Timpani Snapper, demand 10, sell 10: 0
Fish Andalusian Brim, demand 15, sell 15: 0
Fish Plagal Cod, demand 20, sell 20: 0
Fish Fugue Flounder, demand 30, sell 30: 0
Fish Modal Bass, demand 50, sell 50: 0
Paid James, weekly rate=500 amount 6000
Paid rent/utilities 1500
Warehouse Main: Fertiliser cost 2.00
Warehouse Main: Feed cost 400.00
Warehouse Main: Salt cost 200.00
Warehouse Auxilliary: Fertiliser cost 1.00
Warehouse Auxilliary: Feed cost 200.00
Warehouse Auxilliary: Salt cost 100.00
List of Vendors
1. Slippery Lakes
2. Scaly Wholesaler
>>> Enter number of vendor to purchase from: 1
Hatchery Name: Eastaboga, Cash: 1206.03
    Warehouse Main
        Fertiliser, 20.00 (capacity=20)
        Feed, 400.00 (capacity=400)
        Salt, 200.00 (capacity=200)
    Warehouse Auxilliary
        Fertiliser, 10.00 (capacity=10)
        Feed, 200.00 (capacity=200)
        Salt, 100.00 (capacity=100)
    Technicians
        Technician James, weekly rate=500
END OF QUARTER 2
================================
====== SIMULATING quarter 3 ======
================================
To add enter positive, to remove enter negative, no change enter 0.
>>> Enter number of technicians: 0
Fish Clef Fins, demand 25, sell 25: 0
Fish Timpani Snapper, demand 10, sell 10: 0
Fish Andalusian Brim, demand 15, sell 15: 0
Fish Plagal Cod, demand 20, sell 20: 0
Fish Fugue Flounder, demand 30, sell 30: 0
Fish Modal Bass, demand 50, sell 50: 0
Paid James, weekly rate=500 amount 6000
Paid rent/utilities 1500
Warehouse Main: Fertiliser cost 2.00
Warehouse Main: Feed cost 400.00
```

```
Warehouse Main: Salt cost 200.00
Warehouse Auxilliary: Fertiliser cost 1.00
Warehouse Auxilliary: Feed cost 200.00
Warehouse Auxilliary: Salt cost 100.00
List of Vendors
1. Slippery Lakes
2. Scaly Wholesaler
>>> Enter number of vendor to purchase from: 1
Can't restock Fertiliser, insufficient funds, need 2.40 but only have -
7196.97
Went bankrupt restocking warehouse Main in quarter 3
Hatchery Name: Eastaboga, Cash: -7196.97
    Warehouse Main
        Fertiliser, 12.00 (capacity=20)
        Feed, 360.00 (capacity=400)
        Salt, 200.00 (capacity=200)
    Warehouse Auxilliary
        Fertiliser, 6.00 (capacity=10)
        Feed, 180.00 (capacity=200)
        Salt, 100.00 (capacity=100)
    Technicians
        Technician James, weekly rate=500
END OF QUARTER 3
=== FINAL STATE quarter 4 ===
Hatchery Name: Eastaboga, Cash: -7196.97
    Warehouse Main
        Fertiliser, 12.00 (capacity=20)
        Feed, 360.00 (capacity=400)
        Salt, 200.00 (capacity=200)
    Warehouse Auxilliary
        Fertiliser, 6.00 (capacity=10)
        Feed, 180.00 (capacity=200)
        Salt, 100.00 (capacity=100)
    Technicians
        Technician James, weekly rate=500
```

## Part 2: Data Analytics (33.3%)

Jupyter notebook will be used for this part. Your answer for each step must include both your code and your explanation and answers to questions in the markdown cells. Please refer to the information given for markdown cells in each step.

### Step 1: Crawl a real-world dataset (10% of Part 2)

Find an interesting external (remotely hosted) dataset to extract and analyse. Your dataset can be either extracted from APIs and/or via web scraping. ***Reading data from non-external (i.e., local) files is not permitted for this task***. Extracted data must contain **at least 5 columns and 150 rows**. Save your extracted data in a CSV format.

Examples of data sources are:

- Reddit API
- RSS feeds
- NewsAPIs
- YouTube API
- Nasa website

- Foursquare API
- Space Weather
- Echo Nest API
- GHO OData API
- Athena API
- DHS Program API
- OpenAQ API

Note: "pandas.read_table" is not allowed for this task. You may use data from multiple sources and merge them as long as at least one of them is crawled from APIs/ web scraping that satisfies the requirement.

**Explain in the markdown cells**: *Where does the data come from? What are the variables of interest? How was the data scraped/collected?*

### Step 2: Perform data preparation & cleaning (20% of Part 2)

- Load the dataset into a data frame using Pandas
- Handle missing data, if any.
- Handle any outliers or inconsistencies in the data, if any.
- Perform any additional steps to enrich your data (parsing dates, creating additional columns/features, etc.)

**Explain in markdown cells**: *steps to prepare, clean your data, or extract new features*.

### Step 3: Perform exploratory analysis (30% of Part 2)

Explore your data, examples are as follows:

- Compute the mean, sum, range, and other interesting statistics for numeric columns
- Explore distributions of numeric columns using histograms etc.
- Explore the relationship between columns using scatter plots, bar charts, etc.

**Explain in markdown cells**: steps for exploratory data analysis and your observations.

### Step 4: Ask question about your data (30% of Part 2)

- Ask **one interesting question** about your dataset. Your question must be complex enough to have sub-questions. The quality of your question is assessed based on the complexity and interestingness. Trivial and basic questions will not give you full marks.
- Answer the questions either by computing the results using Numpy/Pandas/Scikit-learn or by plotting graphs using Matplotlib. Perform grouping/aggregation wherever necessary. You can use hypothesis testing and basic modelling (such as regression models) to answer the proposed questions.

**Explain in markdown cells:** *Outline in detail your entire analysis. Explain your insights clearly*. You should include a brief description of the intent /interpretation of each plot.

### Step 5: Summarise and write a conclusion (10% of Part 2)

**Explain in markdown cells**:

- Write a summary of what you've learned from the analysis.
- Share ideas for future work on the same topic using other relevant datasets/sources.