



# FIT9136 Algorithms and Programming Foundations in Python

## Assignment 3

Last Updated: 28 September 2023

# Table of Contents

- [1. Key Information](#)
- [2. Context Information](#)
- [3. Implementation Instructions](#)
  - [3.1. The Investor Application](#)
  - [3.2. Extracting Property Information](#)
  - [3.3. Currency Exchange](#)
  - [3.4. Suburb Property Summary](#)
  - [3.5. Average Land Size](#)
  - [3.6. Property Value Distribution](#)
  - [3.7. Sales Trend](#)
  - [3.8. Identifying a Property of a Specific Price in a Suburb](#)
- [4. Do and Do NOT](#)
  - [4.1. Important Notes:](#)
- [5. Submission Requirements](#)
- [6. Academic Integrity](#)
- [7. Marking Guide](#)
- [8. Getting help](#)
  - [8.1. English language skills](#)
  - [8.2. Study skills](#)
  - [8.3. Things are tough right now](#)
  - [8.4. Things in the unit don't make sense](#)
  - [8.5. I don't know what I need](#)

# 1. Key Information

<b>Purpose</b>	<p>This assignment will develop your skills in designing, constructing, and documenting a small Python program according to specific programming standards. This assessment is related to (part of) the following learning outcome (LO):</p> <ul style="list-style-type: none"><li>● <b>LO4:</b> Investigate useful Python packages for scientific computing and data analysis;</li><li>● <b>LO5:</b> Experiment with data manipulation, analysis, and visualisation techniques to formulate business insight.</li></ul>
<b>Your task</b>	<p>This assignment is an individual task where you will write Python code for a simple application whereby you will be developing a property investment management system as per the specification.</p>
<b>Value</b>	<p><b>30%</b> of your total marks for the unit.</p>
<b>Due Date</b>	<p><b>Friday, 20 October 2023, 4:30 PM (AEDT)</b></p>
<b>Submission</b>	<ul style="list-style-type: none"><li>● Via Moodle Assignment Submission.</li><li>● FIT GitLab check-ins will be used to assess the history of development</li><li>● Jplog will be used for similarity checking of all submissions.</li></ul>
<b>Assessment Criteria</b>	<p>The following aspects will be assessed:</p> <ol style="list-style-type: none"><li>1. Program functionality in accordance to the requirements</li><li>2. Code Architecture and Adherence to Python coding standards</li><li>3. The comprehensiveness of documented code and test strategy</li></ol>
<b>Late Penalties</b>	<ul style="list-style-type: none"><li>● 10% deduction per calendar day or part thereof for up to one week</li><li>● Submissions more than 7 calendar days after the due date will receive a mark of zero (0) and no assessment feedback will be provided.</li></ul>
<b>Support Resources</b>	<p>See Moodle Assessment page and Section 8 in this document</p>
<b>Feedback</b>	<p>Feedback will be provided on student work via</p> <ul style="list-style-type: none"><li>● general cohort performance</li><li>● specific student feedback ten working days post submission</li></ul>

## 2. Context Information

You are working as a developer in a property investment company located in Melbourne, Australia. Your company is providing service for customers from all over the world to invest in properties in Melbourne. As your company grows, the management team has discovered that some of the administrative tasks are time-consuming and repetitive. Therefore, as the only developer in this company, your task here is to automate some of these administrative tasks. To test your implemented functionalities, you are provided a CSV file which contains information about the properties sold in some suburbs from 2010 onwards.

## 3. Implementation Instructions

Your implementation must **include a text interface (a text interface is the least requirement) with functionalities** detailed below. Your program will be evaluated based on its clarity, including the provision of clear information and error messages. Your program should handle exceptions properly and handle unexpected cases wherever necessary. The validation of data types and values should be done within each functionality.

**To ensure that your code can be properly evaluated by the teaching team:**

1. Please ensure that your implemented methods use the same names and method signatures (i.e. number and type of input arguments and the type of the return value) as required,
2. Please ensure that your code is properly formatted, such as proper variable naming conventions, consistent indentations, proper line length, etc.,
3. Please ensure that you provide clear and coherent comments on your code to aid with the graders' interpretation of your code,
4. For more details regarding formatting and commenting, see the [PEP 8 Style Guideline](#).

### 3.1. The Investor Application

You are required to use object-oriented programming to implement the text interface and the following functionalities described from 3.2. to 3.8., which allows users to select and perform certain operations. **You should determine the proper number of classes needed instead of including all the methods in one class (for example, you may create a class 1) *SimpleDataAnalyser* to deal with the loading of data as well as the simple analyses ; 2) *DataVisualiser* to deal with the visualisations; and 3) *Investor* to display a menu, ask for and process user inputs, etc.).** Please make sure to create a `main.py` file to run your implemented text interface. You should also provide an instruction file in PDF format, which is no more than two pages in length, to describe the usage of the interface. For each functionality from 3.2 to 3.8, we list a core method as shown below. **Note that for these core methods described below, you should use the same method name and include the same method parameter. The only permitted modification is to add the “`self`” keyword to the methods.**

## 3.2. Extracting Property Information

The `extract_property_info(file_path)` method is responsible for reading the property information from the data file located at the specified `file_path`. This method returns a dataframe.

## 3.3. Currency Exchange

Since many of the customers are from different countries, the `currency_exchange(dataframe, exchange_rate)` method is responsible for transforming the property prices in Australian dollars into the target currency according to the exchange rate. The variable `dataframe` is the one returned in 3.1., and the data type of `exchange_rate` is float. This method returns a NumPy array of transformed prices.

## 3.4. Suburb Property Summary

The purpose of this method is to display the summary of the properties with respect to the number of bedrooms, number of bathrooms and number of parking spaces for a given suburb. Your task is to write a method `suburb_summary(dataframe, suburb)` where `dataframe` is the one returned by the method described in 3.1. and `suburb` is the target suburb to be analysed. Here, if the value of `suburb` is "all", the method should display the summary from all the suburbs. If the value of `suburb` does not exist in the `dataframe`, an error message should be displayed. The summary should at least include the mean values, standard deviation, median, minimum and maximum. This method does not have a return value.

## 3.5. Average Land Size

Your task here is to implement the method `avg_land_size(dataframe, suburb)` where `dataframe` is the one returned by the method described in 3.1. and `suburb` is the target suburb to be analysed (or all suburbs if the value of `suburb` is "all"). This method calculates and returns the average land size in  $m^2$  of properties in the suburb (or None if the suburb does not exist). Note that the invalid land size values should be excluded and that the size units are different for some properties.

## 3.6. Property Value Distribution

Here, you will be visualising the property values as a [histogram](#) so as to easily present to the customers. You are provided a dictionary of key-value pairs where the keys are the names of

the currency and the values are their exchange rate to Australian dollars, which you can use as a local variable within this method:

```
currency_dict = {"AUD": 1, "USD": 0.66, "INR": 54.25, "CNY": 4.72, "JPY": 93.87, "HKD": 5.12, "KRW": 860.92, "GBP": 0.51, "EUR": 0.60, "SGD": 0.88}
```

Your task is to write a method `prop_val_distribution(dataframe, suburb, target_currency)` which:

- Takes the following parameters:
  - A dataframe
  - A suburb value
  - A target currency (whose default value is “AUD”)
- Converts the property value into the target currency if the currency is in the `currency_dict`. In cases where the target currency does not exist in the `currency_dict`, the program needs to inform the user and generate the histogram in **AUD**.
- Produces a graphical representation (i.e., a histogram) of the property value distribution of the specified suburb (or all suburbs if the value of `suburb` is “all”) based on the converted property values and save this graphical representation as a file.
- In cases where the target suburb does not exist in the `dataframe`, the program should inform the user and generate the histogram with **all** suburbs and save the graph as a file.
- Note that the records with missing price values should be excluded.

You will need to invoke methods implemented in previous steps. This method does not have a return value.

### 3.7. Sales Trend

Here, you will need to calculate the number of properties sold in each year and visualise the results as a [line chart](#) so that a sales trend can be easily observed. The line chart should be saved as an image file locally.

Your task is to implement the method `sales_trend(dataframe)` where `dataframe` is the one returned by the method described in 3.1. The method does not have a return value.

### 3.8. Identifying a Property of a Specific Price in a Suburb

Your task here is to implement the `locate_price(target_price, data, target_suburb)` method to find out if a specific `target_price` value is in the list of prices from a specific `suburb` in. The variable `data` is a dataframe loaded by a previous

method. You should filter the data according to the variable `target_suburb` to get the list of prices for the properties in the `target_suburb`. The variable `target_price` is a numeric value. This method first sorts the list of prices using a **reverse insertion sort** (i.e., an insertion sort algorithm that sorts the list in descending order). Then this method searches for the `target_price` value within the reversely sorted list of prices using **recursive binary search**. A boolean value of True or False will be returned by this method to indicate if the `target_price` value can be found in the list of prices filtered for the `target_suburb`.

## 4. Do and Do NOT

Do	Do NOT
<ul style="list-style-type: none"><li>• Maintain appropriate citing and referencing<sup>1</sup>,</li><li>• Get support early from this unit and other services within the university,</li><li>• Apply for special consideration or for extensions<sup>2</sup> early if needed.</li></ul>	<ul style="list-style-type: none"><li>• Leave your assignment in draft mode (assignments in draft mode will not be marked),</li><li>• Submit late (10% daily penalty applies)<sup>3</sup>,</li><li>• Attempt to submit after 7 days of the due date (they will not be accepted), unless you have special consideration.</li></ul>

### 4.1. Important Notes:

- If **any exceptions/errors** happen when running your program and are not properly handled, you will lose 50% of the functionality where the exceptions/errors occur. For example, if the total mark of one functionality is 10 marks and any exception happens when running this functionality which causes the program to terminate, then the maximum mark you can get is 5 instead of 10.
- Add correct validation and output messages to make your code more user-friendly to users.
- For each function listed in 3.2 to 3.8, add code to test each functionality and also the expected behaviour. The test code should be placed in the **main.py** file as comments. For example, if you have created a class named **DataExtractor** where **extract\_property\_info(file\_path)** is an instance method:

```
# de = DataExtractor()
# df = de.extract_property_info("property_information.csv")
# len(df)
# this should output 118771
```

- This is an individual assignment and must be completed on your own. You must attribute the source of any part of your code that you have not written yourself. Please note the section on Academic Integrity in this document.
- The assignment must be done using PyCharm, Python 3.10.
- The Python code for this assignment must be implemented according to the [PEP 8-Style Guide for Python Code](#).

---

<sup>1</sup><https://www.monash.edu/library/help/citing-and-referencing/citing-and-referencing-tutorial>

<sup>2</sup> <https://www.monash.edu/exams/changes/special-consideration>

<sup>3</sup> e.g.: If the original mark was 70/100, submitting 2 days late results in 50/100 (20 mark deduction). This includes weekends and public holidays.



- The allowed libraries are **OS, NumPy, Pandas, Matplotlib** and **SciPy**. You will receive penalties if you use any other libraries. If you are implementing unit tests, you are allowed to use **unittest**.
- For using any built-in functions/methods, students should seek confirmation from the teaching team.
- Commenting on your code is an essential part of the assessment criteria. In addition to inline and function commenting on your code, you should include comments/markdowns at the beginning of your program file which specify your full name, student ID, the creation date, and the last modified date of the program, as well as a high-level description of the program.
- This assignment cannot be completed in a few days and requires students to apply what they learnt each week as we move closer to the submission date. Please remember to show your progress weekly to your tutor.
- You must keep up to date with the Moodle Ed Assignment 3 forum where further clarifications may be posted (this forum is to be treated as your client). If there are any changes to the specification, apart from making announcements on Ed, we will update the new specification on Moodle.
- Please be careful to ensure you do not publicly post anything which includes your reasoning, logic or any part of your work to this forum, as doing so violates Monash plagiarism/collusion rules and has significant academic penalties. Use private posts or email your allocated tutor to raise questions that may reveal part of your reasoning or solution.
- If any aspect of the assignment specifications is unclear or not explicitly addressed, please ensure to verify with the teaching team and request clarification.
- **In this Assessment, you must NOT use generative artificial intelligence (AI) to generate any materials or content in relation to the assessment task.**

## 5. Submission Requirements

The assignment must be submitted by **Friday, 20 October 2023, 4:30 PM (AEDT)**.

Please start working on your assignment as soon as you can.

The following files are to be submitted on Moodle:

- The **.py** files that you created to implement your assignment (i.e., code and documentation). Note that you should properly name these files to make sure their usage is self-explanatory.
- An instruction file that explains the usage of the program. Name the file `ass3_manual_[studentID].pdf`. The instruction file should be coherent and concise, and should not be more than 2 pages in length.

The above files must be compressed to a .zip file named **ass3\_StudentID.zip** and submitted via Moodle. The **.py** files must also have been pushed to your remote repository (at FIT GitLab server) with an appropriate history as you develop your solutions. Please ensure your commit messages are meaningful. You are **NOT** required to push the history of the instruction file to the FIT GitLab server. **DO NOT** push the .zip file.

- Please note **four** pushes is a minimum, in practice we would expect significantly more. All commits must include a meaningful commit message which clearly describes what the particular commit is about.
- No submissions will be accepted via email.
- Please note we **cannot mark any work on the GitLab Server**. Therefore you need to ensure that you submit correctly via Moodle since it is only in this process that you complete the required student declaration, without which work cannot be assessed.
- It is your responsibility to **ENSURE** that the submitted files are the correct files. We strongly recommend after uploading a submission, and prior to actually submitting in Moodle, that you download the submission and double-check its contents.
- Please **carefully** read the documentation under the “**Special Consideration**” and “**Assignment Task Submission**” on the Moodle Assessments page, which covers things such as extensions, correct submission, and resubmission.
- Please note, if you need to resubmit, you cannot depend on your tutors' availability, for this reason, please be **VERY CAREFUL** with your submission. It is strongly recommended that you submit several hours before due to avoid such issues.
- **There are no restrictions on creating extra functions. Do NOT create redundant functions/methods.**
- **Marks will be deducted for any of these requirements that are not strictly complied with.**

- All built-in sorting functions/methods MUST NOT be used (e.g., `sorted()`, `list.sort()`, `(pandas.*).sort_values()`)

## 6. Academic Integrity

Students are expected to be familiar with the [University Academic Integrity Policy](#) and are particularly reminded of the following:

### **Section 1.9:**

Students are responsible for their own good academic practice and must:

- undertake their studies and research responsibly and with honesty and integrity;
- credit the work of others and seek permission to use that work where required;
- not plagiarise, cheat or falsify their work;
- ensure that their work is not falsified;
- not resubmit any assessment they have previously submitted, without the permission of the chief examiner; appropriately acknowledge the work of others;
- take reasonable steps to ensure that other students are unable to copy or misuse their work; and
- be aware of and comply with University regulations, policies and procedures relating to academic integrity.

### and **Section 2.9:**

Unauthorised distribution of course-related materials: Students are not permitted to share, sell or pass on to another person or entity external to Monash:

2.9.1 any course material produced by Monash University (such as lecture slides, lecture recordings, class handouts, assessment requirements, examination questions; excluding Handbook entries) as this is a breach of the Copyright Compliance Policy and such conduct may be a copyright law infringement subject to legal action; or

2.9.2 any course-related material produced by students themselves or other students (such as class notes, past assignments), nor to receive such material, without the permission of the chief examiner. The penalties for breaches of academic misconduct include

- a zero mark for the assessment task
- a zero mark for the unit
- suspension from the course
- exclusion from the University.

Where a penalty or disciplinary action is applied, the outcome is recorded and kept for seven years, or for 15 years if the penalty was excluded.

## 7. Marking Guide

Your work will be marked as per the following:

- **Method Functionalities - 75 Marks**
  - Extracting Property Information - 2 Mark
  - Currency Exchange - 6 Marks
  - Suburb Property Summary - 10 Marks
  - Average Land Size - 10 Marks
  - Property Value Distribution - 18 Marks
  - Sales Trend - 15 Marks
  - Identifying Price Value - 14 Marks
- **OOP Implementation and Main Logic - 25 Marks**
  - Proper design of classes
  - Clear usage manual
  - Clear instructional messages
  - Good alignment to PEP 8 style
  - Detailed test codes
  - Good program logic
- **Penalty - up to 20 marks<sup>4</sup>**
  - Missing requirements listed in Section 4. Do and DO NOT and section 5. Submission Requirements.

Total: 100 marks, recorded as a grade out of 30

---

<sup>4</sup> The penalty here is excluded from the late submission

## 8. Getting help

### 8.1. English language skills

if you don't feel confident with your English.

- Talk to English Connect: <https://www.monash.edu/english-connect>

### 8.2. Study skills

If you feel like you just don't have enough time to do everything you need to, maybe you just need a new approach.

- Talk to a learning skills advisor: <https://www.monash.edu/library/skills/contacts>

### 8.3. Things are tough right now

Everyone needs to talk to someone at some point in their life, no judgement here.

- Talk to a counsellor: <https://www.monash.edu/health/counselling/appointments>  
(friendly, approachable, confidential, free)

### 8.4. Things in the unit don't make sense

Even if you're not quite sure what to ask about, if you're not sure you won't be alone, it's always better to ask.

- Ask in Ed
- Attend a consultation

### 8.5. I don't know what I need

Everyone at Monash University is here to help you. If things are tough now they won't magically get better by themselves. Even if you don't exactly know, come and talk with us and we'll figure it out. We can either help you ourselves or at least point you in the right direction.