COMP0035 Coursework 01 2024 Coursework specification

1. Table of contents

- Introduction
- Coursework specification
  - Getting started
  - General requirements and constraints
  - Section 1: Data exploration and preparation
  - Section 2: Database design and creation
  - Section 3: Tools
  - Section 4: References
- Submission
- Marking
  - Module learning outcomes
  - Mark allocation
  - Grading criteria
- Appendices
  - Code quality
  - Code that does not fully function
  - Guidance on Moodle

Version: 1. 28/09/24

2. Introduction

The aim of the combined coursework in this module is for you to select and apply some of the relevant software development and data science techniques that are used in a typical project lifecycle.

Coursework 1 focuses on data preparation and database design.

Coursework 2 continues from coursework 1, focusing on requirements, application design and testing.

This document specifies coursework 1 which is worth 40% of the assessment marks available for the module. This is an individual coursework.

You will submit a written report; and a repository of code files that combined meet the requirements detailed in this specification.

Aim to make progress each week of first five weeks of the module, in line with module's teaching activities.

3. Coursework specification

3.1. Getting started

1. Select a dataset using the 'group' selection task in Moodle Week 1 (https://moodle.ucl.ac.uk/mod/choicegroup/view.php?id=6089982). Each 'group' option is associated with a data set. 'Group selection' is a Moodle term for the type of task, the coursework is **individual**.
2. Accept a GitHub classroom assignment. This creates the repository. Instructions are also given in Tutorial 1.
    1. Login to GitHub.com.
    2. Click on the GitHub classroom link (https://classroom.github.com/a/zqVIaThf)
    3. Accept the assignment.
    4. If prompted, accept to join the comp0035-ucl organisation.

3. Download the dataset for your group choice and add it to your repository. Use the links in Moodle (Resources > Datasets). For files > 25MB use GitHub large file storage (https://docs.github.com/en/repositories/working-with-files/managing-large-files/about-git-large-file-storage).

3.2. General requirements and constraints

- Compile all written work into a single report in either PDF or Markdown format. Name the document `coursework1`.
- The report supports the code and techniques used in the coursework. It is not an essay, be succinct. There are no word limits.
- Demonstrate regular use of source code control using GitHub. Create the repository using the GitHub classroom assignment. Keep the repository private. Keep the repository in the ucl-comp0035 organisation.
- You must use the data set allocated to you on Moodle.
- This is an individual coursework. Do not collude with other students using the same data set.
- Use of code AI tools is permitted when writing code. UCL recommends using Microsoft Copilot (https://liveuclac.sharepoint.com/sites/Office365/SitePages/Bing-Enterprise-Chat.aspx) using your UCL credentials. This must be stated in the 'References' section.
- Use relevant techniques from the course, or from data science and/or software engineering processes. Provide references for techniques not included in the course material.
- Diagrams can be hand-drawn and scanned. Using software to draw them does not increase marks.

3.3. Section 1: Data exploration and preparation

The purpose of this section is:

- to use python pandas to describe the data set structure and content; and as a result demonstrate that you understand the data set.
- to use python pandas prepare the data for later use in developing applications. The data you prepare will be used in COMP0034 coursework to create charts in a dashboard app.
- to demonstrate that you can write code that is reusable and understood by other developers.
- to demonstrate that you can apply relevant software engineering and data science techniques.

Code quality is also assessed.

Use only Python and `pandas`. `matplotlib` may be used where pandas `DataFrame.plot()` (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.html) is not sufficient.

Create charts where they support your exploration and preparation; but do not focus on the visual aesthetic as this is not assessed.

You may need to prepare the data in order to complete the exploration and hence your code may not neatly split between 1.1 and 1.2. This is OK, the code structure does not need to exactly match the report structure.

3.3.1. Section 1.1 Data exploration

1. Code: Write python code to **explore** and describe the data structure and content. Including, but not limited to, size, attributes and their data types, statistics, distribution of the data, etc. Consider potential data quality issues.
2. Report: Describe the results of your exploration of the data. Do not include the code in the report.

3.3.2. Section 1.2 Data preparation

1. Report: Briefly describe a target audience and state at least 3 questions that they might be interested to explore using the data. This defines the purpose for which you will prepare the data.

2. Code: Write python code to **prepare** the data such that it can be used to try to answer the questions for the audience described in step 1. Aim to have sufficient data, and avoid unnecessary data. The prepared data should be in a format that can be read into one or more pandas dataframes from a file (`.csv` or `.xlsx`). If relevant, address any data quality issues identified in section 1.1.
3. Report: Explain how you ensured the data is relevant for the purpose.
4. Include the original and prepared versions of your data set files in your repository.

3.4. Section 2: Database design and creation

The purpose of this section is:

- to demonstrate that you understand the structure of a relational database and the principles of normalisation by designing an appropriate database and drawing this as an entity relationship diagram (ERD).
- to demonstrate that you can write Python code to create an SQLite database based on the ERD. The database you create can be used in COMP0034 coursework in a data driven web application.

3.4.1. Section 2.1: Database design

Design a relational database that can store the data (based on either the prepared or the raw data set, your choice). Consider normalisation.

Document the design as an Entity Relationship Diagram (ERD) that includes the following details as a minimum:

- table(s)
- attributes in each table
- data type of each attribute
- primary key attribute for each table
- foreign key attribute(s) if relevant
- relationship lines between tables

Include the ERD in your report. An explanation is not required, though you may discuss your normalisation if relevant.

3.4.2. Section 2.2: Database code

Write python code that:

- creates a database structure based on the ERD for an SQLite database file.
- takes the data from the dataset file and saves it to the SQLite database file. Note: do not create a database that requires a server such as MySQL or PostgresSQL.

The quality of the code is assessed.

Use relevant Python packages, i.e. `pandas` and `sqlite3`.

3.5. Section 3: Tools

The purpose of this section is to demonstrate appropriate and effective use of relevant software engineering tools.

3.5.1. Section 3.1 Environment management

Provide relevant files and instructions that allow the marker to set up and run your code in a Python virtual environment. They will use `pip` and `setuptools` with the commands:

```
pip install -r requirements.txt
pip install -e .
```

As a minimum, edit the files that were provided in the starter code of the repository:

- `requirements.txt`: list the packages used in your code
- `pyproject.toml`: provide basic project details and code package location
- `README.md`: provide instructions to install and run your code for the data preparation and the database creation

3.5.2. Section 3.2: Source code control

Add the URL for your repository to the report.

Make regular use of source code control.

3.5.3. Section 3.3: Linting

Use a Python linter to demonstrate how your code meets Python style standards such as PEP8, PEP257.

For example:

- state which Python linter you used.
- provide evidence of the results of running the linter.
- if issues are reported by the linter, address these and then run the linter again and show the results.
- if any issue cannot be addressed, explain why not.

3.6. Section 4: References

Include code references in comments in the code files close to where it is used.

Include all other references, if used, in the report.

3.6.1. Section 4.1 Reference use of AI

State either that you used AI, or state that you did not.

If you used AI, include the details stated in the UCL guidance (https://library-guides.ucl.ac.uk/referencing-plagiarism/acknowledging-AI#s-lg-box-wrapper-19164308).

3.6.2. Section 4.1 Dataset attribution

Comply with any license condition required for your data set (given in the data set link in Moodle > Resources > Data sets).

Each license is different and tells you what has to be cited; e.g. see open government licence v3 (https://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/). Typically, but not always, 'attribution' is required: i.e. include a statement listing who owns the data and its location.

4. Submission

Refer to Moodle > Assessment for the deadline date and time.

Submit your work on Moodle in the assignment submission. The submission states the upload format: `.zip` for the code (and report if in markdown) plus `.pdf` for the report (if not in markdown).

GitHub is **not** an acceptable alternative for submission, though its facility to download the code files as zip may be useful to you.

Make sure all files are in the submission. URLs linking to external files cannot be marked as they could be changed after the submission time. The only exception is where the original data files are too large to upload to Moodle - in this exceptional situation list url(s) to the data files in your report or the `README.md` instead.

Do not include your .venv folder in the zip file, this creates unnecessarily large zip files.

Table: Submission checklist

Section	***Report***	***Code files***
1. Data exploration and preparation	Description and explanation.	Python code to explore/describe the data. Python code to prepare the data. Original data set Prepared dataset
2. Database design and creation	Entity Relationship Diagram (ERD).	Python code to create the database. SQLite database file.
3. Tools	Source code control: URL to GitHub repository Linting evidence.	Environment management: `requirements.txt`, `pyproject.toml`, `README.md`
4. References	Statement of AI use. Data set attribution. Other references if used.	Include code references within the code files.

5. Marking

5.1. Module learning outcomes

The module's published learning outcomes that are assessed in this coursework are indicated in the table.

Learning outcome	*Coursework 1 ?*
1. Describe how software development methodologies can be used to manage the software development process and select and apply an appropriate methodology for a given project.	
2. Select and apply techniques for capturing and modelling requirements.	
3. Select and apply techniques for modelling an application; and model an application using these.	Yes - database
4. Select the aspects of a software application can be modelled with the Unified Modelling Language (UML); and use UML to model different views of an application.	ERD (not UML)
5. Model the design for a database.	Yes
6. Describe testing and recommend an appropriate approach to testing for a given project.	
7. ~~Recognise the challenges of working in a team and organise themselves and their group to deliver a complex project.~~	
8. Recognise the ethical implications of using data in the context of this course and be aware of their responsibilities to comply with relevant UCL and UK legislation.	Yes
9. ~~Work in a group to~~ apply the skills and knowledge gained in the course to: a) produce a coherent and cohesive specification for an application; and b) select, install, configure and use a set of open-source tools and use these to support the software development cycle for the application.	Yes, part (b)

The published learning outcomes are being revised. In particular:

- Data preparation and visualisation are core to the module content yet missing from the published learning outcomes.
- Following feedback from previous students, the coursework is now individual. Learning outcome 7 is not addressed; and learning outcome 9 needs to be re-worded.

5.2. Mark allocation

You are expected to spend 18 hours on coursework 1 (45 * 40%).

The weighting of each section is shown with an indication of the expected hours of effort required.

Section	*Weighting*	*Effort (hours)*
Data preparation and understanding	45%	8
Database design and creation	35%	6
Tools	20%	3

5.3. Grading criteria

The coursework is assessed according to the standards set in the standard UCL Computer Science grading criteria (see copy on Moodle in the Assessment section). The criteria most relevant to this assessment are 1, 2, 4 and 5.

The following tables give the standard UCL CS criteria, and indicators specific to this coursework. The coursework is open-ended and allows for different solutions; it is not possible to describe every aspect that could be considered.

The descriptors typically focus on quality and standard of the response, rather than quantity. If you are going to do something more (quantity), then you are advised to focus on demonstrating something that has not already been evidenced in your work.

5.3.1. Generic CS Descriptors

Grade band	***Generic CS Descriptor***
Distinction 90+	Exceptional response with a convincing, sophisticated argument with precise conclusions. Exceptional grasp of complexities and significance of issues. Exceptional thought and awareness of relevant issues. Sophisticated sense of conceptual framework in context. Exceptional solution and advanced algorithm/technical design.
Distinction 70-89	A distinctive response that develops a clear argument and sensible conclusions, with evidence of nuance. Thorough grasp of issues; some sophisticated insights. Concepts deftly defined and used with some sense of theoretical context. Excellent algorithmic solution, novel and creative approach.
Merit 60-69	A sound response with a reasonable argument and straightforward conclusions, logical conclusions. Sound understanding of issues, with insights into broader implications. Good solution, skilled use of concepts, mostly correct and only minor faults.
High pass 50-59	A reasonable response with a limited sense of argument and partial conclusions. Reasonable grasp of the issues and their broader implications. Reasonable reproduction of ideas from taught materials. Rudimentary definition and use of concepts. Reasonable solution, using basic required concepts, several flaws in implementation.
Low pass 40-49	An indirect response to the task set, towards a relevant argument and conclusions. Rudimentary, intermittent grasp of issues with confusions. Analysis relying on the partial reproduction of ideas from taught materials. Some concepts absent or wrongly used. Rudimentary algorithmic/technical solution, but mostly incomplete.

5.3.2. Data preparation and understanding

Grade band	***Coursework-specific indicators***
Distinction 90+	Evidence beyond the earlier indicators - these solutions are distinctive and as such there is no set indication of what might be included.
Distinction 70-89	Evidence of a thorough understanding of the data. Analysis and preparation is clearly explained and decisions justified in the context of the intended purpose. Code quality is high. Effective code structure. Effective code documentation. Error handling thorough and consistently applied.
Merit 60-69	Evidence of a good understanding of the data. Actions taken that would allow the data set to be used for the intended purpose. Code quality mostly adheres to Python standards. Evidence of structure. Appropriate documentation. Evidence of error handling.
High pass 50-59	Evidence that the student has understood and/or prepared the data using code, though may be more limited. Decisions taken are not clearly explained. The purpose given is appropriate for the data. Reasonable code, using basic required concepts, several flaws in implementation.
Low pass 40-49	The described purpose may not be clear and/or relate well to the given data. Insufficient evidence that the data has been described and explored. Rudimentary preparation code, but mostly incomplete.

5.3.3. Database design and creation

Grade band	***Coursework-specific indicators***
Distinction 90+	Evidence beyond the earlier indicators.
Distinction 70-89	ERD shows an understanding of potential issues that have been considered in the structure and the extent of the normalisation. Code quality is high. Effective code structure. Effective code documentation. Error handling thorough and consistently applied. Data and relationships are correct in the database.
Merit 60-69	ERD is appropriate for the data given its intended use in applications. Design is clear and uses correct notation. Evidence of appropriate normalisation for the intended use. Code quality mostly adheres to Python standards. Evidence of structure. Appropriate documentation. Evidence of error handling. There may be minor issues in the data/ relationships in the database.
High pass 50-59	ERD provided and adheres to notation but may lack minor detail and/or limited evidence of the application of normalisation concepts. Appropriate code that generates a database file with data. There may be minor issues with the code, code quality or data.
Low pass 40-49	ERD provided but may not adhere to an appropriate notation and/or misses required detail. Code shows some understanding but may not generate a usable database file.

5.3.4. Software engineering tools

There are no generic CS criteria relating to this aspect.

Grade band	*Coursework-specific indicators*
Distinction 90+	Near flawless use of a range of appropriate tools. Use of tools beyond the expected tools.
Distinction 70-89	Effective to exceptional use of the expected tools.
Merit 60-69	Appropriate use of the expected tools.
High pass 50-59	Mostly appropriate use of the expected tools.
Low pass 40-49	Limited, or inappropriate, use of the expected tools.

'Regular' use of source code control is stated in section 3.3. 'Regular' cannot be precisely define since students work over different periods. You are expected to make progress on your coursework weekly. Commits over a period of weeks could be considered 'regular'; commits only during a short period such as 1-2 days could not be considered 'regular'.

6. Appendices

6.1. Code quality

This is considered as:

1. Code that is easy for others to read and understand.
2. Code that is re-usable. The focus in this IEP minor is on writing code that could be used in applications, not simply on whether the code works.

When you are writing code consider:

- code structure, e.g. use of functions, classes, modules, packages.
- adherence to python conventions (PEP8 style guide (https://peps.python.org/pep-0008/), PEP275 docstring conventions (https://peps.python.org/pep-0257/)).
- documentation (docstrings, comments).
- error handling.

6.2. Code that does not fully function

If your code does not fully work, and you cannot 'debug' and fix it before submission, then in the relevant section of the coursework document state as much of the following as you can:

- What is the code that doesn't work (e.g. a function name)
- What you think the problem may be. This shows you understand the issue even if you cannot solve it.
- Any solutions you have tried. This shows that you understand the issue and were able to take steps to try and resolve it.

Clear code documentation (docstrings, comments) is often useful in these situations as the marker can more easily see what you intended your code to do, even if it does not fully achieve that.

6.3. Guidance on Moodle

- Referencing: AI, Code, other (https://moodle.ucl.ac.uk/mod/page/view.php?id=6363796)
- Assessment information Q&A: support, late submission, SoRA and EC, submission date change (https://moodle.ucl.ac.uk/mod/page/view.php?id=6363705)
- Computer science grading criteria (https://moodle.ucl.ac.uk/mod/resource/view.php?id=6089928)

- Examples of previous student work (https://moodle.ucl.ac.uk/mod/page/view.php?id=6363498)
- Data sets: ethics, collusion (https://moodle.ucl.ac.uk/mod/page/view.php?id=6370978)
- Academic terms (https://moodle.ucl.ac.uk/mod/page/view.php?id=6700138)