

COM00142M

Department of Computer Science

ADVANCED PROGRAMMING

FORMATIVE ASSESSMENT BRIEF

Author	Dr Dawn Wood
Assessment Type	Formative
Release	Week 2
Submission	Monday of week 5 by 13.00pm (no late submissions)
Feedback	Within 10 working days of submission

I. Module Learning Outcomes

1. Demonstrate a critical understanding of the theory and application of advanced programming techniques
2. **Design and implement programs for real-world problems**
3. **Communicate design decisions for the selection, storage and manipulation of data**
4. Critically evaluate the legal and ethical impact of software developments in real-world contexts.

This formative will contribute to and support the development of a software application on which the summative assessment is based.

II. Assessment Background/Scenario

You should use the client brief and given data set for your formative work. As this brief is identical to the summative you should not share or distribute your work with your peers. Your summative work should be all your own work.

Client Brief

This document provides the client brief which should be used for the development of a single program or a collection of related programs, for submission with the final summative report.

You have been asked to design and develop a **prototype** application that demonstrates how data from the given data set can be formatted, cleaned, and used to generate specific outputs (as listed below).

Functional requirements

The application should provide the following functionality:

- A means to load the initial data set (which consists of three CSV files) and translate it into a suitable format, either XML, or JSON or an entity relationship structure (not CSV)
- A means to back up the data in this format using either files or a database. This should preserve the current state of the data when the program is closed, and make it available when the program is reopened.
- A process for cleaning and preparing the initial data set, managing inconsistencies, errors, missing values and any specific changes required by the client (see below).
- A graphical user interface(s) for interacting with the data that enables the user to:
 - Load and clean an initial data set (from the CV format)
 - Load and save a prepared data set (from its translated format)
 - Use the prepared data set to generate output and visualisations
 - Manipulate the range of values used to generate output and visualisations

It should be assumed that this program will be able to handle other sets of data generated from the same source, i.e. data with the same column row headings but containing different values and anomalies. However, the application is **not** required to be generic (work with multiple unknown data sets). Given this best practice regarding code reuse, encapsulation and a well-defined programming interface should be applied where applicable.

Data manipulation and outputs

The client initially wants the application to perform the following actions on the data:

1. Outputs should not include any data from airports that have a **'type' 'closed'**
2. The **'type'** column contains information of the type of airport. Extract this out into a new column, one for each category of airport, for:
 - a. all UK(GB) airports, that are , **large_airport, medium_airport, small airport**
 - b. join each category, **large_airport, medium_airport, small airport** to the communication frequencies **'frequency_mhz'** that the airport uses for communication ensuring that each airport in all categories is correctly matched with its communication frequencies.
3. The client initially needs information to generate the following and output the results using appropriate representation:
 - a. Produce the mean, mode and median for the **'frequency_mhz'**
 - i. For each **large_airport**
 - ii. For frequencies **more than 100 mhz**
4. Produce a suitable graph that display the communication frequencies used by 'small_airport' You may need to consider how you group this data to make visualisation feasible
5. Determine if there is any significant correlation between the communication frequencies used by the 3 different categories of airport. 'Are some frequencies used more than others?'. You will need to select an appropriate visualisation to demonstrate this.

Non-functional requirements

- The GUI interface provides appropriate feedback to confirm or deny a user's actions
- The application manages internal and user-generated errors

Technical requirements

- The application is built using Python 3.7.*
- The application uses one or more of the advanced APIs introduced on this module such as: NumPy, panda, Seaborn, Matplotlib. It should NOT use alternative APIs for this functionality, however Python core libraries can be used to support where applicable, such as support for a database.
- The application runs within the anaconda environment using a Jupyter notebook
- The application or its parts do not run concurrently, do NOT use Python threads

The requirements specified here are the constraints within which you need to produce your development. They are not negotiable with the client.

III. Assessment Task(s)

You should use the above brief and the production opportunities at the end of each week to develop either a single piece of software or a collection of working programs. These will form the necessary learning and development to enable you to undertake and provide appropriate evidence in your summative assessment report.

The formative submission provides you with an opportunity to receive feedback on your approach and thinking process, which will prepare you for the summative. The summative assessment is based 100% on the report you produce (**NOT** the production of the software), which requires you to evidence your development with appropriate code samples, design documents and justifications on your approach and decisions. Given this the formative will focus on providing feedback to support this, NOT on correcting code samples.

Given the three productions from week 2-4 (inclusive), select **ONE** to submit as your Formative work. Make it clear which production you are submitting. The following is a copy of the information you will find on each week's *production* page on Canvas.

Week2, Production 1: Data translation for storage

To enable a program to function each time it runs there needs to be an external and persistent data storage system that retains the state of the data. There are two considerations here. The physical storage medium for the data, such as a file or database and the format/structure of the data such as CSV or XML. This week you have explored a number of different formats in both regards as follows:

- CSV
- XML
- JSON
- MongoDB
- SQL Database
- Other file formats?

Select **ONE** format that you consider as most suited to the data in the scenario and the aims of the program. The format selected should support both the nature of the data and the aims of

the application being designed. It should provide distinct advantages and minimal limitations over other data formats. It should not be selected solely because it is the easiest to program, although this can be included as an advantage if applicable.

Design:

Produce a model that shows how the data needs to be restructured to take best advantage of the selected format and work more effectively within the program. Where you have created groups or objects from the data show how they relate to each other.

Implementation:

Implement a parser that reads in the original data file. You may want to create a subset of the data file for testing and speed. Your program should then perform the translation from the original format/structure into your selected format. The result of this process should then be outputted to its relevant physical medium (files/database).

At this stage there is no requirement to handle data types (other than those inherent in the data format, i.e. numbers and “Strings”), conversions or missing data. The program can be demonstrated as a simple console based application, requiring the input of the file name by the user and sufficient output to demonstrate the correctness of the translation process.

Your program should produce regular output statements to the console so that it is easy to follow what the program is doing and provides a visual demonstration of the translation process. This will also be handy for any debugging required.

Reflection on design decisions:

Write a 200-word reflection that states the reason for your format selection and the advantages the format leads to the data and application and any limitations on the future use of this data within the selected format. You should consider what literature supports your reasoning/decisions.

Week 3, Production 2: Interactive GUI

Given the client brief, there are a number of requirements to develop a GUI to enable users to interact with the data and generate statistical information and visualisations. For this production focus on designing the layout and identifying the types of interaction that will be required.

Design:

Produce a set of wireframe designs for the application. These do not need to be final, changes can be made after the formative submissions, however they should focus on giving a complete view of the applications interface (not a collection of different ideas for one part of the interface). It should be clear how one part of the interface leads to another, and where any additional windows are opened.

Alongside this produce a set of integration diagrams (state-machines are a possible option here) to demonstrate 2 or 3 key areas of interaction. It should be clear which aspects of the interface these interactions relate too.

Implementation:

Create a first iteration implementation of the interface design. This may include small adaptations that are different from the design but you should aim for at least 90% of the design being similar. That is to say it should be very clear that one (design) is a model of the other (implementation). Avoid designing via coding, you will find the result disjointed. The key to GUI is design on paper first build second.

Reflection on design decisions:

Write a 200-word reflection that states the reason for the selection of the layout and components for the interfaces design. Clearly identify which specific aspects of the requirements or data's structure informed the decisions. You should consider what literature supports your reasoning/decisions

Week 4, Production 3: Data Cleaning and Initial Analysis

Given the client brief, there are a number of requirements to provide accesses to specific parts of the data and provide answers to specific statistical questions. For this production focus on how your application will manipulate the data (cleaning and shaping) and developing functions for calculating some of the statically requirements.

Design:

Consider the steps required for cleaning and shaping and any of the calculations (functions) you want to develop. Write pseudocode to sketch these out before you write code. Mentally or on paper walk the data through your pseudocode steps to test how effective your solution is.

Implementation:

The first stage is to clean the data and make sure it is fit for purpose. Examine the data carefully to identify anomalies and then consider how your program can identify these and correct/delete or change. You will need to consider how you are going to handle erroneous or missing values. You should output a sample of the data that demonstrates how cleaning has changed the data.

The next stage is to reshape the data as per any requirements of the brief. Is all the data needed to provide the required results? Is any of it duplicated? Is there data across different sources that needs to be brought together? Again, output a sample to the console to demonstrate how this has changed the structure of the data.

Finally develop and test a set of functions (or objects and methods) that applies the statistical analysis to the data set, outputting the results to the console.

Capture the results of your data cleaning, shaping and functions with screenshots of the console. There is no requirement at this stage for anything to be functioning in through the GUI. Make sure it is clear what your output is testing/demonstrating (output simple informative statements).

Reflection on design decisions:

Write a 200-word reflection that states why you have selected specific tools from NumPy or pandas. This may be the data structure you have used, the functions you have applied to clean and shape the data. Clearly identify which specific aspects of the requirements or data's structure informed the decisions. You should consider what literature supports your reasoning/decisions.

NOTE: If you submit more than one production, **only the first** will receive formative feedback. If you combine productions, making it difficult to distinguish between them, only the first 4 A4 pages will be considered

IV. Deliverables

You should make **two file** submissions as follows:

- A design reflection and all designs/models/images as a single file in either .docx or .pdf format. This should NOT be included in the zipped file and should not exceed any specified word/page counts.
- A single zipped file containing your program or programs.

Submission guidelines:

- Submit **ONE** production only
- Follow the standard guidelines on Canvas for **Submission Formatting**.
- Your reflection/justification **should not exceed 200 words** and should be presented in at least 12pt font size.
- You should submit **no more than 4 * A4 pages** in total which includes your reflection, designs, screenshots, or other relevant images.
- Your code should be focused on the selected production and not the whole program, and be easily identifiable as a solution to the production problem. Well commented and structured code is more likely to be easy to follow and receive feedback.

V. Formative feedback

Formative feedback will be given on the following areas and support the development of the learning outcomes indicated:

Learning Outcome	Section/Task	Criteria
2	Design:	Are the design tools and techniques selective effective? Do the designs convey the right level of information given their context? Is the design appropriate in the context of the client's brief?
3	Implementation:	Are the appropriate code constructs/functions/APIs selected? Are there sufficient comments and output statements (when the program runs) to convey what the program is doing? Is the program logical in its decomposition and approach?
3	Reflection:	Does the reflection clearly capture the thought process, and justify the decisions underpinning the development? Are these decisions supported by the documentation, program, and relevant literature?

VI. Grading - None

No grades are awarded for formative assignments. Feedback will indicate your general level of ability to tackle the task and present your ideas and arguments, and your adherence to academic best practice, in regards to Master's level awards.

VII. Assessment submission

You will submit your assessments in the 'Assignments' area of the module in Canvas. Please check your Canvas module for the specific submission date for this assignment.

Formative work **MUST** be submitted by the deadline, there is no LATE submission available.

For general assessment guidelines consult your Canvas Module, Orientation Module and for Academic Regulations the University of York's website.

Any queries regarding the details of your assessment should be directed to your module tutor.

Any queries regarding assessment procedures should be directed to studentsuccess@online.york.ac.uk.