COM00142M

Department of Computer Science

Advanced Programming

SUMMATIVE ASSESSMENT BRIEF

|  |  |
| --- | --- |
| **Author** | Mr Tony Knowles |
| **Assessment type** | Summative assignment |
| **Weighting** | 100% |
| **Release** | Week 3 |
| **Deadline** | Monday following Week 8, 13:00 (UK time) \* |

\* If this date falls on a UK public holiday or a University of York closure day, the submission date will change. Please check the submission point in the ‘Assignments’ area of the module in Canvas for the exact submission deadline.

# Module Learning Outcomes

The module learning outcomes for this module are as follows:

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 assessment addresses **all** the module learning outcomes listed above

# 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 suitable 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 DAB Radio stations that have the following **‘NGR’** : NZ02553847, SE213515, NT05399374 and NT252675908
2. The **‘EID’** column contains information of the DAB multiplex block E.g C19A. Extract this out into a new column, one for each of the following DAB multiplexes:
   1. all DAB multiplexes, that are , **C18A**, **C18F**, **C188**
   2. join each category, **C18A**, **C18F**, **C188**

to the **‘ NGR’** that signifies the DAB stations location to the following: **‘Site’,** ‘**Site Height, In-Use Ae Ht, In-Use ERP Total**

* 1. Please note that: **In-Use Ae Ht, In-Use ERP Total**  will need the following new header after extraction: Aerial height(m), Power(kW) respectively.

1. The client initially needs information to generate the following and output the results using appropriate representation:
   1. Produce the mean, mode and median for the **‘In-Use ERP Total’** from the extracted DAB multiplexes extracted earlier: **C18A**, **C18F**, **C188**
      1. For ‘**Site Height’** more than 75
      2. For **‘Date’** from 2001 onwards
2. Produce a suitable graph that display the following information from the three DAB multiplexes that you extracted earlier: **C18A**, **C18F**, **C188:**  **‘Site’, ‘Freq’, ‘Block’, ‘Serv Label1’, ‘Serv Label2’, ‘Serv Label3’, ‘Serv label4’,’Serv Label10’**

You may need to consider how you group this data to make visualisation feasible

1. Determine if there is any significant correlation between the **‘Freq’, ‘Block’, ‘Serv Label1’, ‘Serv Label2’, ‘Serv Label3’, ‘Serv label4’,’Serv Label10’** used by the extracted DAB stations. You will need to select an appropriate visualisation to demonstrate this.

### Non-functional requirements

* The GUI interface must be able to provide appropriate feedback to confirm or deny a user’s actions
* The application must be able to manage internal and user-generated errors

### Technical requirements

* The application is built using Python 3.7 - 3.10
* The application uses one or more of the advanced APIs introduced on this module such as: NumPy, pandas, 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.

# Assessment Task(s)

Given the client brief above, produce a suitable software solution that meets the specified requirements as either a single program, or a series of clearly identifiable programs.

**NOTE:** Failure to submit the program(s) will result in a zero grade.

From this development produce a structured design report (using the given template) that addresses a series of focus questions. These ask you to discuss your design decisions and demonstrate the underpinning theoretical aspects in the context of your software development. Code samples should be extracted from your software development where requested to demonstrate specific algorithms and interactions. You should support your discussion with appropriate reference to relevant sources using the correct citation and reference structure as indicated in the guide to [IEEE referencing system](https://www.york.ac.uk/students/studying/develop-your-skills/study-skills/study/integrity/referencing-styles/ieee/).

## Report contains 3 sections as follows:

The report consists of three sections, each containing specific questions targeting what needs to be addressed. In each section is a maximum word count and this should be adhered to. Your report should clearly identify which question you are addressing. Ensure you keep your code samples separate from your explanations. The sections contribute to the module learning outcomes as follows:

**Section 1:** Theory supported by code samples (50%, 1400 words plus code samples)

Evidence for learning outcome: Demonstrate critical understanding of the theory and application of advanced programming techniques; Design and implement programs for real world problems.

**Section 2:** Design decisions supported by code samples (40%, 1200 words plus code samples)

Evidence for learning outcome: Communicate design decisions for the selection, storage and manipulation of data; Design and implement programs for real world problems.

**Section 3:** Reflection on the ethics, moral and legal aspects (10%, 400 words)

Evidence for learning outcome: Critically evaluate the legal and ethical impact of software developments within real world contexts.

# Deliverables

Your assignment should be laid out following the formatting guidelines that are specified in the ‘Submission Formatting’ page in Canvas.

You should make two file submissions as follows:

* A completed report answering the given questions 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. If a database has been used you should produce a file dump of the table structure to include here. This should NOT contain the original data set.

**NOTE:** Failure to submit the program(s) will result in a zero grade.

The report must adhere to the word count limits stated for EACH question and the page limit for the appendices.

## Code samples:

All code samples submitted in your report should be extracted from your program verbatim. Those that are not will not contribute towards your grade. Code samples do **NOT** contribute to your word count. You should make sure they very clearly target what is requested, and submitting general or broad ranges of code may be subject to penalties. For example, if you were asked to demonstrate where a closure has been used within your code, you would not be demonstrating this by submitting a complete program that contains a closure. You may add code style comments to your code samples to help target what you are showing, but should not change the structure of the code from your software submission.

## Using a database:

Where you have opted to use an SQL or relational database (other than Mongo) include after your list of references the following:

* Name of database and link to download (install package)
* Version number of the database used
* The name of the code file (Python class/Jupyter Notebook) that creates and populates the database
* The point in your code where local host and the port are set (make this clear)

You should make sure that your submitted code contains all the code required to setup and populate your database via a local host connection.

## General submission guidelines:

* Present your answers on A4 pages, with a minimum 12 point font (14 point for headings), minimum 120% line spacing (what Word calls “Multiple 1.08”), and margins of at least 2 cm on either side.
* All images must be of a size/quality that the details are viewable within the document. Images will not be scaled to identify blurred or small text.
* 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.

## Referencing

You are required to use the [IEEE referencing style](https://subjectguides.york.ac.uk/referencing-style-guides/ieee) for citing books, articles, and all other sources (such as websites) used in your assignment.

Good referencing is essential in order to meet the standards of academic integrity set by the University. All your sources must be acknowledged, regardless of whether you included direct quotes or not. Visit your **Academic Integrity Tutorial** module in Canvas for additional guidance on effective referencing.

# Marking Criteria

Students should be able to use this as a guide to the elements they need to cover, without it giving them an answer to the task/question. Where academic quality is required there should be a separate and clear criterion for this (e.g. 10%). The criteria should be clearly linked to the module level learning outcomes (via Number) this can be at the task or element level.

|  |  |  |  |
| --- | --- | --- | --- |
| **Learning Outcome** | **Section/Task** | **Criteria** | **Available marks** |
| 2 | Upload code and design documentation | **Pass/fail** | **0** |
| Section 1. Theory supported by code samples | | | |
| 1,2 | a. Adaptation to a concurrent model | Appropriate concurrent mechanisms have been selected, which are discussed in terms of the advantages and disadvantages to the client brief and the developed program. | **20** |
| 1,2 | b. Implementing user interaction | The selected constructs are appropriate given the required interactions. There is a clear rationale for their selection given best practice in GUI constructs and layout. | **10** |
| Appropriate code samples and designs are given that demonstrate the overall effectiveness of the user interaction given the client’s requirements. | **10** |
| 1,2 | c. Evaluating high level languages | There is a rational argument, either for or against, Java or Python being more effective in terms of ‘speed of development’, that is supported by a critical evaluation of each language’s approach and constructs, which is supported by appropriate referencing | **10** |
| Section 2. Design decisions supported by code | | | |
| 2,3 | a. Data format | An effective format has been selected and a rational argument is presented for how it supports the nature of the data and the type of analysis required to produce the client’s information needs. | **10** |
| 2,3 | b. Code constructs | Appropriate code constructs and internal data structures have been selected and applied to achieve the client’s requirements. Considerations have been made for any anomalies within the data set. | **10** |
| 2,3 | c. Data visualisation | Appropriate and effective libraries/functions have been selected and their application for presenting the required output is logical. Considerations have been made for any anomalies within the data set. | **10** |
| 2,3 | d. Data analysis | Appropriate and effective libraries/functions have been selected and their application for preparing the data is logical. Considerations have been made for any anomalies within the data set. | **10** |
| Section 3. Reflection on ethics, morals and legal aspects | | | |
| 4 | a. Reflection | There are clear and appropriate examples, which are effectively used to support either a ‘for’ or ‘against’ position on the statement. | **10** |
|  | TOTAL | | **100** |

# Marking Criteria: Grade breakdown

## Question 1a – Adaptation to Concurrent Model [20%]

|  |  |  |
| --- | --- | --- |
| 0-39% | Fail | No mechanisms identified, or mechanisms are inappropriate. No reference to the client brief. |
| 40-49% | Compensatable fail | Mechanisms identified are not justified. Very limited discussion, passing or limited reference to the client brief. |
| 50%-59% | Pass | Mechanisms are appropriate and justified. |
| 60%-69% | Merit | As Pass + reference to wider reading, literature used to justify choices. |
| 70%-100% | Distinction | As Merit + in-depth or very strong argument for mechanisms. References are wide. |

## Question 1b – Implementing User Interaction [20%] This question has 2 parts: GUI Constructs and Appropriate Samples – each element is worth 10%. 1b GUI Constructs [10%]

|  |  |  |
| --- | --- | --- |
| 0-39% | Fail | Constructs are not appropriate or there is no rationale. |
| 40-49% | Compensatable fail | Constructs are appropriate but rationale for selection/use is weak. |
| 50%-59% | Pass | Constructs are appropriate for the brief, rationale for selection/use is present. |
| 60%-69% | Merit | As Pass + reference to wider reading, literature used to justify choices. |
| 70%-100% | Distinction | As Merit + in-depth or very strong argument. References are wide. |

## 1b Appropriate Samples [10%]

|  |  |  |
| --- | --- | --- |
| 0-39% | Fail | Code sample/design not present, inappropriate, or not effective. |
| 40-49% | Compensatable fail | Code samples are weak, or the design is not clear. |
| 50%-59% | Pass | Code sample/design is clear and shows how the brief is met. |
| 60%-69% | Merit | As Pass + demonstrates effectiveness. |
| 70%-100% | Distinction | As Merit + clear and obviously effective sample/design. Highly effective/"best practice". |

## 1c Evaluating High Level Languages [10%]

|  |  |  |
| --- | --- | --- |
| 0-39% | Fail | Descriptive, no evaluation or comparison. No supporting Python examples. No reference to language-specific approaches or constructs. |
| 40-49% | Compensatable fail | Mainly descriptive, the argument is weak or flawed. Limited evaluation and Python examples from the student’s own code: |
| 50%-59% | Pass | Arguments are present with valid justification. Supported by reference to Python samples from the student’s own code and evidence |
| 60%-69% | Merit | As Pass + evaluation is obviously critical with some clear interaction with the evidence being used from a selection of other sources beyond the prescribed reading. |
| 70%-100% | Distinction | As Merit + strong, clear, fluent argument that has depth and shows insight, supported by wider literature that is used to support analysis and evaluation and isn’t just used for context setting. |

## 2a Selected Data Format [10%]

|  |  |  |
| --- | --- | --- |
| 0-39% | Fail | Selected format is flawed. No argument provided. |
| 40-49% | Compensatable fail | Format is present and workable, but the argument is limited, flawed, or mainly descriptive. |
| 50%-59% | Pass | Format is appropriate to the argument presented. That argument is not flawed and refers to the client brief. |
| 60%-69% | Merit | As Pass + format is effective, supported by reference to literature. |
| 70%-100% | Distinction | As Merit + format is highly appropriate or is in line with "best practice"; argument is strong, clear, fluent and well supported. |

2b Code Constructs [10%]

|  |  |  |
| --- | --- | --- |
| 0-39% | Fail | Code is not appropriate, or has obvious major issues/flaws. Raw data set used, or data set not cleaned or processed. No discussion or justification. |
| 40-49% | Compensatable Fail | Code is somewhat appropriate in some places. Some processing performed, though it may not be effective/purposeful. Discussion is descriptive. |
| 50%-59% | Pass | Code is appropriate. Processing is effective or otherwise justified. Constructs are discussed and/or justified. |
| 60%-69% | Merit | Code is effective in cleaning and processing data. Constructs are justified with reference to literature. |
| 70%-100% | Distinction | Code is highly effective or in line with "best practice". Justification is strong, clear, fluent and well supported. |

2c Data Visualisation [10%]

|  |  |  |
| --- | --- | --- |
| 0-39% | Fail | Visualisations not present or are not appropriate. Little or no discussion of choices. |
| 40-49% | Compensatable Fail | Visualisations are present, but not all are appropriate or complete. The discussion is descriptive. |
| 50%-59% | Pass | Visualisations are appropriate and the choice is discussed/justified. |
| 60%-69% | Merit | Visualisations are effective and justified with reference to literature. |
| 70%-100% | Distinction | Visualisations are clear and highly effective. Use of wider reading to justify and support choices. |

2d Data Analysis [10%]

|  |  |  |
| --- | --- | --- |
| 0-39% | Fail | Data analysis is not present or is only very limited. |
| 40-49% | Compensatable Fail | Data analysis is present, but limited or incomplete, and discussion is descriptive only. |
| 50%-59% | Pass | Analysis is appropriate, correlation is discussed and linked to analysis. Conclusions are reasonable. |
| 60%-69% | Merit | Analysis is effective, justification includes reference to wider reading. |
| 70%-100% | Distinction | Analysis is insightful and well supported by wider literature. |

3a Ethics, Moral & Legal [10%]

|  |  |  |
| --- | --- | --- |
| 0-39% | Fail | Discussion not present or very limited. |
| 40-49% | Compensatable Fail | Appropriate examples given but with limited or descriptive discussion, or discussion of inappropriate examples. |
| 50%-59% | Pass | Examples are used appropriately to support an evaluation of the ethical, moral and legal aspects of the question, with some support from wider reading. |
| 60%-69% | Merit | As Pass. Evaluation is obviously critical, with frequent reference to and support from wider reading. |
| 70%-100% | Distinction | As Merit. Critical evaluation is insightful, clear and fluent. Wider reading is used to directly support the argument throughout. |

# Assessment Submission

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

This assessment requires you to anonymously upload your submission to Canvas. If you are submitting multiple files, you must upload all files simultaneously to ensure that they are marked as a single submission. If you want to resubmit one component of your work, you need to re-upload all other files at the same time: every submission must include **all** files required by the assessment brief.

We recommend that you allow at least 30 minutes before the deadline to upload your submission, as failure to upload your assessment file within the allotted time is not admissible as an exceptional circumstance.

The webpage [How do I submit an online assignment?](https://community.canvaslms.com/t5/Student-Guide/How-do-I-submit-an-online-assignment/ta-p/503) provides further technical information on how to upload an assessment. The advice given here comes directly from Canvas. We do not recommend uploading assignments by mobile. We recommend you view the submission after uploading your work to ensure the correct file has been submitted and no technical errors have occurred.

If you face any technical difficulties whilst trying to submit this assessment, then contact Canvas support on [support@instructure.com](mailto:support@instructure.com) or +44 80 0060 8442 (available 24 hours) in advance of the deadline. You should also email [cs-online-admin@york.ac.uk](mailto:cs-online-admin@york.ac.uk) as a matter of urgency to report the issue and receive further instruction.

# Assessment Policies

This assessment is subject to the policies stated on the ‘Summative Assessment Policies’ page in Canvas. These policies include (but are not limited to):

* Academic Integrity and submission of student work to Turnitin
* Advice on anonymising your assessment
* Penalties for late submission
* Marking policy for multiple submissions
* The Fit to Sit / Submit policy
* Passing mark and module reassessment

Please ensure that you have read and understood these policies before starting the assessment.