Assignment Guidance and Front Sheet

This front sheet for assignments is designed to contain the brief, the submission instructions, and the actual student submission for any WMG assignment. As a result the sheet is completed by several people over time, and is therefore split up into sections explaining who completes what information and when. Yellow highlighted text indicates examples or further explanation of what is requested, and the highlight and instructions should be removed as you populate ‘your’ section.

This sheet is only to be used for components of assessment worth more than 3 CATS (e.g. for a 15 credit module, weighted more than 20%; or for a 10 credit module, weighted more than 30%).

**To be completed by the student(s) prior to final submission:**

Your actual submission should be written at the end of this cover sheet file, or attached with the cover sheet at the front if drafted in a separate file, program or application.

|  |  |
| --- | --- |
| **Student ID or IDs for group work** | **e.g. 1234567** |

**To be completed (highlighted parts only) by the programme administration after approval and prior to issuing of the assessment; to be consulted by the student(s) so that you know how and when to submit:**

|  |  |
| --- | --- |
| **Date set** | 2nd October 2023 |
| **Submission date (excluding extensions)** | 19 February 2024 |
| **Submission guidance** | The submission link can be found on Tabula. |
| **Marks return date (excluding extensions)** | 18th March 2024 |
| **Late submission policy** | If work is submitted late, penalties will be applied at the rate of **5 marks per University working day** after the due date, up to a **maximum of 10 working days** late. After this period the mark for the work will be reduced to 0 (which is the maximum penalty). “Late” means **after the submission deadline time as well as the date** – work submitted after the given time even on the same day is counted as 1 day late. |
| **Resubmission policy** | If you fail this assignment or module, please be aware that the University allows students to remedy such failure (within certain limits). Decisions to authorise such resubmissions are made by Exam Boards. Normally these will be issued at specific times of the year, depending on your programme of study. More information can be found from your programme office if you are concerned.  **If this is already a resubmission attempt, this means you will not be eligible for an additional attempt. The University allows as standard a maximum of two attempts on any assessment (i.e. only one resubmission). Students can only have a third attempt under exceptional circumstances via a Mitigating Circumstances Panel decision.** |

**To be completed by the module owner/tutor prior to approval and issuing of the assessment; to be consulted by the student(s) so that you understand the assignment brief, its context within the module, and any specific criteria and advice from the tutor:**

|  |  |
| --- | --- |
| **Module title & code** | Smart Solutions Development I (Programming), WM164-30 |
| **Module owner** | Dr Jianhua Yang |
| **Module tutor** | See above |
| **Assessment type** | Project - Individual |
| **Weighting of mark** | 60% |

|  |
| --- |
| **Assessment brief** |
| For this assignment, you are tasked with designing a Python GUI program to address various analytics questions using given data. It's important to emphasize that the design of your program and the code's style are as crucial as your answers to the analytics questions. A comprehensive report for your program is also required. Refer to the "**Marking Guidelines**" section for details on how the report will be evaluated.  **The Dataset**  You must choose from one of the two provided dataset options.  **Option 1:**  During the C++ part of the module, we developed Arduino-based air quality monitoring kits, which were then distributed to local schools: Cardinal Newman Catholic Secondary School and Sixth Form College in Coventry, and Harris Church of England Academy in Rugby. The collected and cleaned data is now available on Moodle. The data, stored in two CSV files, comprises continuously monitored Air Quality Indicators (AQIs) from two distinct locations. Each record in the file includes a timestamp and AQI readings.  **Option 2:**  This is the Bristol air quality dataset originally obtained from data.gov.uk[[1]](#footnote-1). The version on Moodle, specifically for this assignment, is a simplified edition containing AQI readings from only two locations. The data is stored in a single CSV file. Similar to Option 1, each record contains a timestamp and two AQI readings corresponding to different locations.  It's noteworthy that in both options, some data points might be missing, and your program should be robust enough to identify and handle such scenarios.  **Functional Requirements**  You are required to address the following questions, referred to as functional requirements (FRs). All FRs must be addressed, meaning you must create a GUI to present these analytical outcomes.  **FR1**: Calculate the mean AQI readings' differences between the two locations.  **FR2**: Determine, within a 3-point time window, at which location/time the window has the highest standard deviation. 3-point time window means three adjacent rows that AQIs are measured as illustrated in the figure below.  **FR3**: Display analytical results of FR1 and FR2 in a user-friendly GUI. Output presented on the command line will not be considered, so be sure to use the GUI to display analytical results.    **Constraints**  You must utilize the provided data as-is and exclusively as input to your program. Altering the given data or embedding it directly in your code is prohibited.  You are limited to using Python version ≥ 3.7.  While you can use external software libraries beyond the Python Standard Library, these libraries' source code must be freely accessible online, accessible to assessors and examiners. If uncertain, verify with the module tutor.  **The Report**  Your report should be a single file in either Word or PDF format. Specific points to note:   * Include the word count of your report on the front page. * Provide the URL to your project repository on the WMG GitLab server (https://mygit.wmg.warwick.ac.uk) on the front page. Do not use public servers such as GitHub. * Include all Python source code in report appendices. * Your report should be comprehensible to individuals with general technical backgrounds, even if they are not coding experts. * While you can use appendices for additional information, ensure the main text contains all essential information needed to understand the project. * While report structure is flexible, adhering to the suggested structure and aligning with marking criteria is recommended: * Introduction: Including general project overview, GUI and FRs. * Coding Principles: Evidence for LO7, including coding philosophy (test-driven, data-driven), and more. * Object-Oriented Programming (OOP): Evidence for LO4, including encapsulation, inheritance, and more. * Quality and Style: Evidence for LO6, including adherence to PEP8, naming conventions, and more. * Version Control: Evidence for LO5, including usage of Git features, rationale, justification, and more. * Conclusions: Summarizing key findings, reflections, future prospects. * Appendices: Source code to replicate claimed results, additional information if necessary.   **Best Practices**   * Use an academic writing style; avoid first-person language (I/we). * Effectively utilize illustrations (tables, figures) with clear labels and captions. Reference them in your text. * Support your design decisions and claims withappropriate references**.** * Incorporate important syntax-highlighted code snippets in the main body of the report; avoid extensive code blocks. * Do not submit source code files separately. |

|  |  |
| --- | --- |
| **Word count** | **3000 words + 10%**  You must declare the word count on the front page of your report, not providing a word count is penalised by a reduction in marks of 5%.  The word count includes titles, headings, captions, abstracts & summaries, quotations, text in tables, footnotes, endnotes, in-text citations. It does not include text in figures, the questions as set, or other material from the rubric or supplementary information such as tables of contents, tables of figures, tables of tables, list of acronyms, glossary, code snippets, the reference list, or bibliographies. Material submitted in an annex or appendix is also outside the word count. However, all materials maybe considered in the assessment process and are within marking scope and so are subject to academic integrity rules.  Note: use of text in image format or any non-searchable form in the report to defeat word count or Turnitin analysis is an academic integrity offence and will be reported as such.  The deductions for over-length work are as follows:  For more than 10% up to and including 30% a deduction of 10 percentage points will be applied. For more than 30% , the work will be capped at the pass grade (marks below pass grade are returned without penalties). |
| **Module learning outcomes (numbered)** | LO1 Describe basic syntax, data structures, and algorithms used in modern programming languages.  LO2 Describe development techniques and tools used in modern programming languages.  LO3 Describe the underlying benefits of memory management and code optimization, and how to interact with basic hardware devices.  LO4 Describe underlying theories of object-oriented programming, and how these are applied at different stages of software development.  LO5 Describe underlying theories of version controls systems.  LO6 Describe factors affecting software quality and approaches for how to control them including code quality and standards.  LO7 Describe the basic principles of coding, testing and debugging software for data analysis involving different data formats. |
| **Learning outcomes assessed in this assessment (numbered)** | LO 4/5/6/7 |
| **Marking guidelines** | Refer to file ‘DTS WM164 23\_24 Assignment UG Mark Sheet.docx’. |
| **Academic guidance resources** |  |

1. <https://www.data.gov.uk/dataset/3f5d7bbf-73af-46cd-8b9e-8e07cedb9506/bristol-air-quality>, accessed 18th August 2023. [↑](#footnote-ref-1)