Nottingham Trent University

School of Science and Technology

COURSEWORK ASSESSMENT SPECIFICATION

Module Code	SOFT40161
Module Title	Introduction to Computer Programming
Module Leader	Dr M. Arif Rahman
Module Team	Dan Buxton
Coursework Title	**Python Programming for Real World Data Analytics and GUI Development**
Module Learning outcomes assessed	All
Apprentice Learner KSBs evidenced	K1, K2, S1, S2, and S3
Contribution to module (include contribution to element if appropriate)	100% of the Module
Date work set	10.10.2024
Deadline for submissions	**17.01.2025; 2.30 pm**
Method of Submission	Now Dropbox, One .ipynb and one .pdf file (Same Content; different file format)
Deadline for Feedback	By 3 Weeks after the deadline
Method of Feedback	NOW Dropbox
Previous coursework that may support this submission.	N/A

Late submissions and NECs

Work handed in up to five working days late will be given a maximum Grade of Low Third/ Pass whilst work that arrives more than five working days will be given a mark of Zero.

Please note if you are repeating the work and are capped in your grade, you will receive a Zero grade if the work is submitted at all late.

Work will only be accepted beyond the five working day deadline if satisfactory evidence, for example, an NEC is provided. https://www.ntu.ac.uk/studenthub/my-course/student-handbook/submit-a-notification-of-extenuating-circumstances

Presentations (delete as appropriate)

As a student you may have a statement of access concerning aspects of presentations such as eye contact or reading from notes. While these aspects of presentation will be part of the GBA grid used to assess work, it will not affect your grade.

Breaches of Academic Integrity

To ensure that you are not accused of any breaches of Academic Integrity, look at the NOW page Plagiarism and Academic Integrity at NTU. for guidance.

The University views **plagiarism and collusion** as serious academic irregularities and there are a number of different penalties which may be applied to such offences. The Quality handbook a section on such breaches, which outlines the penalties and states that **plagiarism** includes:
Presenting someone else's ideas as your own (**including text, graph, diagrams, videos etc.**) in a substantial proportion of your work, with or without consent, by incorporating it into assessment without full acknowledgement, including:
Self-plagiarism: reproducing or representing work for assessment without proper attribution and attempting to gain credit for this work where credit has already been received.
Paraphrasing: rephrasing a source's ideas without proper attribution
Mosaic plagiarism/patchworking: weaving phrases and text from several sources into your own work; and/or adjusting sentences without quotation marks or attribution.
Source-based plagiarism: providing inaccurate or incomplete information about sources such that they cannot be found.
Computer code plagiarism: copying or adapting source code without permission from and attribution to the original creator.

Whereas **collusion** includes:
working with other students on an assessment meant for individual submission Sharing your work with other students enabling them to plagiarise your ideas.

Please remember submitting portions of work already assessed for the same learning outcomes is **Self-Plagiarism** and is also a serious academic irregularity.

Penalties for Breaches of academic integrity range from capped or zero grades for elements of modules, to dismissal from the course and termination of studies.

Chat GPT and other AI-powered language models.

It is important to note when using any AI platform that they generate the most common responses to questions, not necessarily the correct ones. They also fabricate evidence. The material they produce is not your own words. Assessments require you answer questions giving your own view and in your own words. The outputs from platforms such as Chat GPT do not provide that.

By presenting such material as your own words you are violating Academic Integrity policy, a matter that NTU takes very seriously.

The skills you develop during your time with us allow you to interrogate material and evaluate it, important skills in all careers. Generative AI does not allow you to develop these.

If you have utilised such platforms, you must retain any outputs from them to provide evidence your work is your own in the case of suspected breaches of Academic Integrity

I. Assessment Requirements

Section 1: Control Structures (16 marks)

***Description:* Implement control structures (if statements, loops etc.) to analyse data.**

Task: Select a publicly available dataset (e.g., from Kaggle, UCI Machine Learning Repository, or government databases) related to a real-world problem (e.g., climate change, public health, or economic indicators. An example of datasets is given at the end). Write code within a Jupyter Notebook that uses control structures to process the data, deriving insights such as trends, averages, or anomalies based on specific conditions.

Deliverables:

Jupyter Notebook (.ipynb): Submit a well-documented Jupyter Notebook with clear code, explanations, and analysis using control structures. Include markdown cells as part of the written report to describe the dataset, methods, and key findings. Ensure all results are displayed within the notebook (Find the given .ipnyb Template from NOW).

Written Report: The markdown cells within the Jupyter Notebook will summarise the dataset and real-world problem. Explain how control structures were used, provide key findings from the analysis, and include a reflection on results and limitations.

Code Quality: Ensure the code is clean, readable, and follows best practices. Use meaningful variable names, appropriate indentation, and clear logic, adhering to Python coding standards.

Section 2: Functions and Modules (16 marks)

***Description:* Design and implement functions and use modules effectively.**

Task: In the same Jupyter Notebook, create multiple functions to handle different aspects of data analysis for the selected dataset. Additionally, import and utilize at least 3 or 4 external modules (e.g., `requests` for API calls, `NumPy` for numerical operations) to enhance the program's capabilities. Ensure the functions are reusable, well-documented, and modular.

Deliverables:

Jupyter Notebook (.ipynb): Use the same notebook as section 1, expanded to include multiple well-structured functions for different aspects of data analysis. Use external modules effectively, ensuring proper imports and clear integration into the analysis process. The functions should be documented within markdown cells to explain their purpose and usage.

Written Report: The markdown cells in the notebook will also serve as the report, detailing the functions, the rationale for modularizing the code, and the use of external modules. Provide explanations for the functions created, the role of each module, and how they contribute to the overall analysis.

Code Quality: Maintain high code quality, with attention to modularity, readability, and reusability. Ensure appropriate commenting, meaningful variable names, and adherence to Python coding standards throughout.

3. Section 3: Data Handling with Pandas (16 marks)

Description: **Use Pandas for data manipulation and analysis.**

Task: Analyse the dataset using Pandas within the Jupyter Notebook. Perform tasks such as data cleaning (handling missing values and duplicates), filtering, and aggregating data to derive meaningful insights. Document all steps clearly using markdown and comments to explain the logic and methodology.

Deliverables:

Jupyter Notebook (.ipynb): Continue using the same notebook to implement data manipulation and analysis using Pandas. Include tasks like data cleaning, filtering, and aggregating to extract insights. Document the logic and workflow in markdown cells, ensuring clarity in the steps taken for each analysis.

Written Report: The markdown documentation within the notebook serves as the report. Provide detailed explanations of the data handling process, including how missing values, duplicates, and other inconsistencies were managed. Discuss the results of filtering and aggregation, linking them to the overall analysis objectives.

Code Quality: Ensure code adheres to good coding practices with organised and readable Pandas operations. Comment clearly, and use descriptive variable names to enhance understanding of the data processing steps.

4. Section 4: Data Visualization (16 marks)

Description: **Create visual representations of data and develop a simple graphical user interface (GUI).**

Data Visualization: Using Matplotlib or Seaborn, generate at least three different types of plots to visualise trends or relationships within the dataset. Incorporate customisations such as subplots or interactive plots (using Plotly or Seaborn). Ensure that the visualisations include appropriate titles, labels, legends, and annotations with an inclusive choice of colors.

Deliverables:

Jupyter Notebook (.ipynb): In the Jupyter Notebook (.ipynb), you will use Matplotlib or Seaborn to create at least three different types of plots to visualize trends or relationships within the dataset. It is essential to document your code clearly in markdown cells, explaining each visualization and the logic behind the choice of plots, as well as any customizations made, such as colors, labels, and legends. Each plot should include appropriate titles, axis labels, legends, and annotations where necessary, ensuring an inclusive choice of colors for accessibility.

Written Report: The markdown documentation within the notebook will serve as the report, providing detailed explanations of the visualization process. You should discuss how each plot contributes to understanding the dataset and justify the choices of colors and customizations that enhance clarity and accessibility. Additionally, explain any insights gained from the visualizations and how they relate to the overall objectives of the analysis.

Code Quality: For code quality, ensure that your code adheres to good coding practices by focusing on organized and readable plot creation. Include clear comments throughout the code to explain the function of each part, and use descriptive variable names to enhance understanding of the visualization steps and choices.

5. Section 5: GUI Development (16 marks)

***Description:* Create visual representations of data and develop a simple graphical user interface (GUI).**

GUI Development (16 marks): Create a simple Python GUI using libraries like Tkinter or PyQt to allow users to interact with the dataset. For example, the GUI might allow users to select specific data subsets or perform basic analysis tasks (e.g., displaying summary statistics, generating specific visualizations) by interacting with buttons or input fields. Focus on usability and the ability to perform core functions within the GUI.

Deliverables:

Jupyter Notebook (.ipynb): The Jupyter Notebook (.ipynb) will include the code for the GUI development, demonstrating how users can engage with the dataset. You should provide thorough documentation in markdown cells that explain your design choices, the functionality of the GUI, and any usability considerations that were taken into account during development.

Written Report: The markdown documentation within the notebook will serve as the report, detailing how the GUI facilitates user interaction with the dataset. You should explain the core functions available in the GUI and highlight the usability features that enhance the overall user experience.

Code Quality: To ensure code quality, your implementation should adhere to good coding practices. This includes an organized structure, clear comments throughout the code, and the use of descriptive variable names. Following guidelines for creating intuitive user interfaces will further enhance the usability and effectiveness of your GUI.

6. Section 6: Version Control, Critical Appraisal, Documentation, and Report (20 marks)

Description: Demonstrate the use of version control practically, prepare a comprehensive report, and document the code clearly.

Version Control: Set up a GitHub repository for the coursework and commit work incrementally throughout the project. Ensure:

- Regular commits with clear messages describing changes.

- A well-structured repository (e.g., directories for data, notebooks, and documentation).

Overall Reporting: Within the same Jupyter Notebook (that you have created earlier), include:

- Introduction: Brief overview of the dataset and real-world problem at the beginning.

- Implementation/Reporting: Description of data processing and analysis techniques (Sections 1 to 5 individually as mentioned in the deliverables for individual sections).

- Findings: Key insights and visualisations (Sections 1 to 5 individually as mentioned in the deliverables for individual sections).

- Conclusion: Summary of results and implications at the end.

Code Documentation: Ensure that each section of the code is thoroughly commented to explain the purpose and logic. Use Python's docstring format for documenting functions.

Coding Standards and Critical Appraisal: Follow consistent coding practices and code design. Possible methods of critical appraisals are-

- *Code Reviews*- Peer review can help identify issues and improve code quality.
- *Static Analysis Tools*- Automated tools can detect potential problems like syntax errors, code smells, and security vulnerabilities.
- *Unit Testing*- Writing tests for individual code units can help verify correctness and identify regressions.
- *Profiling*- Analysing the performance of the code can help identify bottlenecks and areas for optimisation.

Reference: Follow Standard Referencing.

Sample Dataset:

I shall expect you will find your own choice and explore that dataset accordingly. I am adding one example here just for your understanding.

Global Land and Surface Temperature Trends:

This dataset from Kaggle contains global land and surface temperature data from major cities around the world. By relying on the raw temperature reports that form the foundation of their averaging system, researchers are able to accurately track climate change over time. With this dataset, we can observe monthly averages and create detailed gridded temperature fields to analyse localized data on a country-by-country basis. The information in this dataset has allowed us to gain a better understanding of our changing planet and how certain regions are being impacted more than others by climate change. With such insights, we can look towards developing better responses and strategies as our temperatures continue to increase over time

Link: https://www.kaggle.com/datasets/thedevastator/global-land-and-surface-temperature-trends-analy/data

Submission Requirements

1. A single Jupyter Notebook file NTUID_XXXXXX.ipynb and also the exported NTUID_XXXXXX.pdf of the same notebook file containing all code, visualisations, and written explanations, with clear documentation for others to easily understand the code and it's rationale.
2. A link to the GitHub repository demonstrating version control usage.

II. Assessment Criteria (contextualised GBA Grid)

SOFT40161: Introduction to Computer Programming Coursework marking sheet						
Student:	Student ID:	Marker: Dr Arif Rahman and Dan Buxton	Grade:	ENTER GRADE	Grade points:	0.0
General Feedback:						
General Feedback:						

Criteria	Distinction				Commendation			Pass			Fail			ZERO	Grade points
	DEXC	DHIGH	DMID	DLOW	CHIGH	CMID	CLOW	PHIGH	PMID	PLOW	FMARG	FMID	FLOW		
	16	15	14	13	12	11	10	9	8	7	6	4	2	0	
Section 1: Control Structures (16 marks) Implementation of control structures to analyze data and generate insights	Control structures are used in an advanced, efficient manner. The logic is well-thought-out, demonstrating strong problem-solving skills. Code is clean, thoroughly documented, and follows coding standards.				Control structures are well-implemented with mostly efficient logic. Some room for optimization. Code is adequately documented, with minor inconsistencies.			Control structures are functional but may be inefficient or redundant. Code documentation is basic but present, and some coding standards are followed.			Control structures are present but contain significant logical flaws or inefficiencies. Code lacks proper documentation and shows inconsistent adherence to coding standards.			Control structures are missing or incorrectly implemented. No documentation provided. Little to no adherence to coding standards.	
Comments															
Section 2: Functions and Modules (16 marks) Design and implementation of functions and use of external modules	Functions are modular, reusable, and demonstrate an advanced understanding of Python. External modules are used in creative and effective ways. Code is well-structured and fully documented.				Functions are well-designed and generally reusable. External modules are used effectively but without much innovation. Code follows good practices, but some documentation may be incomplete.			Functions are implemented but lack modularity or clarity. External modules are used but not integrated effectively. Code has minimal documentation, with some coding standard issues.			Functions and modules are present but poorly implemented, lacking reusability. External modules are used incorrectly or sparingly. Code documentation is weak.			Functions and external modules are either missing or incorrectly used. Code lacks documentation and does not follow coding standards.	
Comments															
Section 3: Data Handling with Pandas (16 marks) Manipulation, analysis, and	Advanced Pandas features are used for complex data manipulation tasks (cleaning, filtering, aggregation) with clear and efficient code. Excellent documentation, adhering to all coding				Pandas is used effectively for data handling, though more advanced features could be utilized. Documentation is adequate, with minor gaps in code structure or			Basic Pandas operations are used, but the code lacks efficiency or complexity. Documentation is minimal but present. Some coding standards are followed.			Pandas is used, but the implementation is inefficient or lacks clarity. Documentation is			Little to no use of Pandas for data handling. Incorrect or incomplete implementation. No documentation provided, and no coding standards	

handling of data using Pandas	standards.	explanation.		poor, and coding standards are not consistently followed.	followed.	
Comments						
Section 4: Data Visualization (16 marks) Visual representation of data using Python libraries.	Visualizations are highly informative, well-customized, and demonstrate an advanced understanding of data presentation. Multiple plot types are used effectively, with full customization (subplots, interactive elements). Code is well-documented.	Visualizations are effective but lack full customization. The use of multiple plot types is adequate, though some may lack clarity. Code documentation is generally good, with minor gaps.	Basic visualizations are provided but lack depth or customization. Limited variety in plot types. Code documentation is minimal but present.	Visualizations are unclear or ineffective in presenting data. Minimal customization. Documentation is poor.	Visualizations are missing, incomplete, or incorrect. No customization and no documentation provided.	
Comments						
Section 5: GUI Development (16 marks) Development of a GUI for data interaction and analysis	The GUI is highly functional, user-friendly, and allows for complex interaction with the dataset. Code is modular and fully documented, with a clear focus on usability and performance.	The GUI is functional and allows for basic interaction with the dataset. Some usability issues are present, and code is documented but may lack some clarity.	The GUI is basic and lacks interactivity or usability. Documentation is minimal, and the code does not consistently follow best practices.	The GUI is incomplete, poorly designed, or lacks functionality. Minimal or no documentation provided.	The GUI is missing or completely non-functional. No documentation provided.	
Comments						
Section 6: Version Control, Critical Appraisal, Documentation, and Report (20 marks) Use of GitHub, comprehensive report, and code documentation	Version control is used consistently with detailed and meaningful commit messages. The report is clear, thorough, and well-organized, documenting the problem, methodology, and findings in depth. Critical Appraisal is with excellent standard. Code is fully documented and follows all coding standards.	Version control is used effectively, with regular commits and good commit messages. The report is comprehensive, though some details may be lacking. Critical Appraisal is with good standard. Code documentation is good but could be improved.	Basic version control is in place, but commits are infrequent or lack detail. The report is sufficient but lacks depth in methodology or findings. Critical Appraisal is partially there but inconsistent. Code documentation is minimal.	Version control is used sporadically or incorrectly. The report is incomplete or poorly structured, with unclear methodology and findings. Little to no code documentation or critical appraisal.	No or incorrect use of version control. The report is missing or severely incomplete. Code lacks documentation and fails to meet coding standards and critical appraisal.	
Comments						

Nottingham Trent University

III. Feedback Opportunities

Formative (Whilst you're working on the coursework)

Summative (After you've submitted the coursework)

Is

IV. Referencing styles:

General Reference guidance
Referencing in the style of Harvard
Referencing in the style of Vancouver.
Referencing in the style of Science and Justice Journal
Referencing in the style of IEEE
Referencing in the style of the Royal Society of Chemistry
Guide to planning your time here and an automated planner here
Remember to use Outlook or physical calendars to block out time between lectures and labs to work on this coursework.

V. Moderation

The Moderation Process

All assessments are subject to a two-stage moderation process. Firstly, any details related to the assessment (e.g., clarity of information and the assessment criteria) are considered by an independent person (usually a member of the module team). Secondly, the grades awarded are considered by the module team to check for consistency and fairness across the cohort for the piece of work submitted.

VI. Aspects for Professional Development

The skills acquired through this coursework are immensely valuable and highly transferable for employment applications across various fields. Implementing functions and using external modules enhances your programming skills, showcasing your ability to write reusable and modular code, which is essential in software development. Mastering data manipulation with Pandas equips you with the ability to analyse and derive insights from complex datasets, which is a critical competency in data-driven roles. The experience in data visualisation using Matplotlib and Seaborn enhances your ability to communicate findings effectively and demonstrates creativity and analytical thinking.

Creating a graphical user interface (GUI) with Python showcases your software development skills, emphasising your ability to design user-friendly applications that facilitate data interaction. Additionally, the knowledge of version control through GitHub underscores your commitment to collaborative work and project management, demonstrating the ability to maintain organised, well-structured repositories with regular commits and clear documentation.

These competencies illustrate adaptability, attention to detail, and effective communication, qualities employers highly value. Whether in data analysis, software development, or project management, these transferable skills position you as a strong candidate in today's competitive job market.