

## Solent University Coursework Assessment Brief

### Assessment Details

Module Title:	Programming for Problem Solving
Module Code:	COM728
Module Leader:	Jarutas Andritsch
Level:	7
Assessment Title:	The Software Project
Assessment Number:	AE1
Assessment Type:	Software Artefact and documentation
Restrictions on Time/Word Count:	Documentation No more than 2000 words (excluding Table of contents, Table of Figures, Index of Tables)
Consequence of not meeting time/word count limit:	It is essential that assignments keep within the time/word count limit stated above. Any work beyond the maximum time/word length permitted will be disregarded and not accounted for in the final grade.
Individual/Group:	Individual
If a group	-
Assessment Weighting:	60%
Issue Date:	25 <sup>th</sup> September 2023
Hand In Date:	9 <sup>th</sup> January 2024 by 4:00 pm.
Planned Feedback Date:	Within 4 working weeks
Mode of Submission:	Online via SOL <b>Only FINAL submissions will be accepted. DRAFT submissions will not be considered an attempt and will not be marked.</b>
Number of copies to be submitted:	<ul style="list-style-type: none"> <li>- 1 copy of a zip file containing the source codes: main program as <b>Jupyter notebook (.ipynb)</b>, own-defined <b>module as .py</b> and <b>dataset for your software artefact</b></li> <li>- 1 copy of a software document in PDF format. This should <b>not</b> be included in the zip file but instead submitted as a separate file.</li> </ul>
Anonymous Marking	This assessment <b>is exempt from anonymous marking.</b>

### Assessment Task

You are required to develop a software application that addresses the problem scenario using Python and the tools specified in this assessment brief. You must document software implementation which provide a concise and critical discussion of your solution. You should discuss how your solution has been implemented with suitable justifications.

### Introduction

**PhoneDB** website, which is the information website for smartphones, tablets, PDAs, and mobile devices. PhoneDB offers a comprehensive collection of **data and various services** that harness the potential of this valuable resource to aid users in finding the most suitable mobile device.

In this assessment, you will **process, manage, and analyse device features** extracted from the PhoneDB website. You will work with this real-world data set which is provided to you in the form of a **CSV file**. The data file, **device\_features.csv**, contains **48 columns**. Each row in the file represents a single record for a device. The data set contains complete data for all columns for each record in the file. This means that there are no missing values. Some columns may contain **multiple values within the same column**. It is recommended that you familiarise yourself with the content of the data file before attempting the remainder of this assessment.

## Requirements

The requirements for the system are as follows:

- a) The system will allow the user to retrieve data from a CSV file using the csv module and fundamental python (control structure and file processing) to perform the following:
  - **Load the data from a CSV file into memory using the csv reader function**. The path to the file will be specified by the user **then use these loaded data to perform following tasks**:
    - a1. Retrieve the model name, manufacturer, weight, price, and price unit for the device(s) based on the oem\_id.
    - a2. Retrieve the brand, model name, RAM capacity, market regions, and the date when the information was added for device(s) associated with a specified code name.
    - a3. Retrieve the oem\_id, release date, announcement date, dimensions, and device category of the device(s) based on a specified RAM capacity.
    - a4. Retrieve information from your chosen columns and apply a specific condition that relates to an individual device. Please select at least three columns and one condition that differs from previous requirements.
- b) The system will allow the user to **analyse/query data using** the **pandas** module to perform the following:
  - Load data from a CSV file into memory using the pandas module. Use the file path received from task a) for this purpose. After loading the data, proceed with the following tasks.
    - b1. Identify the top 5 regions where a **specific band** of devices was sold.
    - b2. Analyse the average price of devices within **a specific band, all in the same currency**.
    - b3. Analyse the average mass for each manufacturer and display the list of average mass for all manufacturers.
    - b4**. Analyse the data to derive meaningful insights based on **your unique selection**, distinct from the previous requirements.
- c) The system will allow the user to visualise the data using the **matplotlib module** as follows:
  - Load data from a CSV file into memory. Use the file path received from task a) for this purpose. After loading the data, proceed with the following tasks.
    - c1. Create a chart to visually represent the proportion of RAM types for devices in the **current market**.
    - c2. Create a chart to visually compare the number of devices for each USB connector type
    - c3. Create separate charts illustrating the **monthly** average price trends (in GBP) for devices released in each year from 2020 to 2023. Each chart should focus on a specific year.
    - c4**. Create a visualisation of **your selection** to showcase information related to device features that can reveal trends, behaviours, or patterns, ensuring it is distinct from previous requirements.

## Software documentation

The software documentation conveys a comprehensive grasp of software implementation while furnishing substantiating evidence pertaining to the identification of legal, social, ethical, and

professional aspects associated with software development. This documentation should encompass the following key topics:

- Overview: the aim and objectives of the project and brief discussion of the dataset
- Self-reported requirement completion
- Project Implementation: Project Structure and self-created module/functions (technically explain how the module/function implemented)

### Expectations

**The assessment must be completed individually. You must not share, in part or whole, your assessment with another party other than the module tutor and for the purpose of submission to the university. You must ensure that the University's academic misconduct guidelines are followed in their entirety.**

It is expected that you will develop a software application that meets the stated requirements. You have been provided with a CSV file that contains data. Your application will need to appropriately load the data contained in this file, process the loaded data, query the loaded data, and visualise suitable information from these loaded data. You should appropriately test your implemented functionality.

You are required to evidence your work throughout your assessment. You should create a suitable **private Git repository**. Please ensure you **regularly commit** your implementation to your repository **with clear and descriptive commit messages** as you work on your solution. As part of the project and demonstration for this assessment, you will be asked to document and show your project's repository. You should ensure that your submission complies with academic misconduct guidelines, is your own work and any external sources have been appropriately referenced. **Failure to provide a Git commit history or unsuitable commit history will result in a potential cap on your overall mark, limiting it to a passing grade.**

Note: If you have any special requirement or disability, please discuss this with your tutor.

### Environment

You are required to use the following tools:

- Jupyter Notebook as your development environment
- Python3.9 or above as the standard python library
- Additionally, the following libraries/modules/function may be imported and utilised:
  - o os – to retrieve or check file paths
  - o random – to generate random numbers
- The COM728 environment from the class session.
- Git Tools and GitHub for version control

**No other python libraries or modules should be used other than the specified.**

## Assessment criteria

Learning Outcomes	UPPER FIRST A1 – A2 Exceed expectations in many aspects		FIRST A3 – A4  Substantially exceeds expectations		UPPER SECOND B1 – B3 (High) Meet learning outcomes and exceeds expectations in several aspects			LOWER SECOND C1 - C3 (Good) Meet learning outcomes and sometimes exceeds expectations			THIRD D1 - D3 (Competent) Meet learning outcomes			FAIL F1 – F3 (Incomplete/Poor) Fails to meet learning outcomes		
SOLENT GRADE	A1	A2	A3	A4	B1	B2	B3	C1	C2	C3	D1	D2	D3	F1	F2	F3
Design computer programs in a logical and structured way using appropriate techniques and principles	The text-based user interface is exceptionally well-structured as a standalone module, seamlessly integrated into the main program and other modules for displaying results. Usability is a central focus, with user interactions driven by meticulously designed structures and functions. It presents results flawlessly with clear, intuitive prompts, and offers insightful error messages when necessary, ensuring an unparalleled smooth and user-friendly experience.		The text-based user interface is expertly developed as a standalone module, effectively integrated into the main program and other modules. Usability is a key priority, featuring well-organised structures and functions for user interactions. It consistently presents results with clear prompts and provides helpful error messages, contributing to a highly smooth and user-friendly experience.		The text-based user interface is competently crafted as a standalone module, successfully integrated into the main program and other modules. Usability considerations are evident, with well-organised structures and functions for user interactions. It generally presents results with clarity and offers helpful error messages, resulting in a mostly smooth and user-friendly experience.			The text-based user interface is competently developed as a standalone module, although integration into the main program and other modules could be improved. Usability aspects are somewhat addressed, with modestly organised structures and functions for user interactions. It typically presents results and offers basic error messages, resulting in an acceptable but not fully polished user experience.			The system utilises a basic user interaction message, usability considerations are minimal keeping all the code within a single file. It presents results plainly in a straightforward layout or format. There are instances where static data is employed in the coding process to meet specific requirements.			No user interface, no user interaction message has been implemented. All attempt coding use static data or initialise data		Usability aspects are largely overlooked, resulting in poorly structured structures and functions for user interactions. Results may lack clarity, and error messages are often unhelpful, leading to a frustrating and suboptimal user experience.

Develop computer programs aligned to appropriate programming standards and code conventions	The software artefacts unequivocally demonstrate flawless realisation of all problem scenario requirements, achieving a perfect 100% correctness and containing no errors. The code is exceptionally well-modularised, expertly combining user-defined and built-in functions that operate precisely as intended and adhere to established best practices. Clear and meticulously structured comments are seamlessly integrated throughout the code, greatly enhancing overall readability. This clarity is further enhanced through steadfast	The software artefacts exhibit an excellent level of proficiency, confidently addressing all problem scenario requirements (90-100%) with excellent correctness, only minor errors. The code demonstrates excellent modularity, effectively incorporating both user-defined and built-in functions, aligned with best practices. Comments are consistently clear and well-structured, contributing to excellent code readability. Coding standards are consistently upheld, with impeccable indentation and uniform	The software artefacts showcase a high level of competence, effectively implementing a significant portion of the problem scenario requirements (80-90%) with high correctness, only occasional errors. The code is highly modular, effectively utilising both user-defined and built-in functions in adherence to best practices. Comments are consistently clear and well-structured, enhancing overall readability. Coding standards are diligently followed, with good indentation and uniform variable/function naming.	The software artefacts display proficiency in addressing problem scenario requirements (60-80%), maintaining correctness, minor errors. The code exhibits modularity, predominantly employing user-defined functions. Comments are generally clear, contributing to code readability. Coding standards are consistently followed, including proper indentation and consistent variable/function naming.	The software artefacts competently execute a majority of the problem scenario requirements (50-60%), with satisfactory correctness, occasional errors. Code modularity is evident, with a preference for user-defined functions. Comments, while existent, may require improvement in terms of clarity and consistency. Coding standards are sporadically adhered to, impacting code readability to some extent.	The software artefacts exhibit limited proficiency in implementing select problem scenario requirements (30-50%), with noticeable correctness issues and errors. Code modularity is limited, predominantly relying on built-in functions. Comments, if present, may lack clarity and consistency, hindering code understanding. Coding standards are inconsistently followed, diminishing code readability.	The software artefacts demonstrate limited implementation of problem scenario requirements (less than 30%), with substantial correctness challenges and frequent errors. Code lacks modularity, primarily relying on built-in functions. Comments, if available, may lack clarity and consistency, making it difficult to understand the code. Coding standards are largely ignored, severely impacting code readability.
---	--	---	--	---	--	---	---

	adherence to coding rules and conventions, including impeccable indentation and consistent variable/function naming.	variable/function naming.					
Utilise suitable tools to design, implement, test and evaluate solutions	Extensive error exception handling and validation have been used in the code	Well error exception handling and validation have been used in the code	There is a good evidence of error exception handling and validation	There is some evidence of error handling or validation but no exception handling	There is a few evidence of error handling	There is no evidence of error or exception handling	No evidence of attempting required threshold

## Learning Outcomes

This assessment will enable you to demonstrate in full or in part your fulfilment of the following learning outcomes identified in the Module Descriptor:

## Living CV

As part of the University's Work Ready, Future Ready strategy, you will be expected to build a professional, Living CV as you successfully engage and pass each module of your degree.

The Living CV outputs evidenced on completion of this assessment are:

1. I can solve real-world problems by getting and analysing large amounts of data.
2. I can confidently write Python code to obtain, manipulate, and analyse real-world dataset.
3. I am experienced in using environment tool such as Jupyter notebook to design, implement, test and evaluate solutions.
4. I can conduct written and verbal presentations to share insights to audiences of varying levels of technical sophistication.

Please add these to your CV via the Living CV builder platform on Solent Futures Online [Solent Futures Online](#)

## Important Information

[Solent University Academic Regulations 2023-24](#)

## Late Submissions

You are reminded that:

- i. If this assessment is submitted late i.e. within 7 calendar days of the submission deadline, the mark will be capped at 40% if a pass mark is achieved;
- ii. If this assessment is submitted later than 7 calendar days after the submission deadline, the work will be regarded as a non-submission and will be awarded a zero;
- iii. If this assessment is being submitted as a referred piece of work, then it must be submitted by the deadline date; any Refer assessment submitted late will be regarded as a non-submission and will be awarded a zero.

## Assessment regulations

### Extenuating Circumstances

The University's Extenuating Circumstances (EC) procedure is in place if there are genuine short term exceptional circumstances that may prevent you submitting an assessment. You are able to self-certify for up to two assessment dates in any semester without supporting evidence for an extension of up to seven calendar days for coursework or to defer an exam to the resit period.

Alternatively, if you are not 'fit to study' (or you have used up your two self-certification opportunities), you can request:

- an extension to the submission deadline of 7 calendar days, or
- a request to submit the assessment at the next opportunity, i.e. the resit period (as a Defer without capping of the grade).

In both instances you must submit an EC application with relevant evidence. If accepted under the university regulations there will be no academic penalty for late submission or non-submission

dependent on what is requested. You are reminded that EC covers only short-term issues (20 working days) and that if you experience longer term matters that impact on your learning then you must contact the Student Hub for advice.

Please find a link to the EC policy below:

#### Extenuating Circumstances

#### **Academic Misconduct**

Any submission must be your own work and, where facts or ideas have been used from other sources, these sources must be appropriately referenced. The University's Academic Regulations includes the definitions of all practices that will be deemed to constitute academic misconduct. You should check this link before submitting your work.

Procedures relating to student academic misconduct are given below:

#### Academic Misconduct

#### **Ethics Policy**

The work being carried out must be in compliance with the university Ethics Policy. Where there is an ethical issue, as specified within the Ethics Policy, then you will need an ethics release or ethics approval prior to the start of the project.

The Ethics Policy is contained within Section 2S of the Academic Handbook:

#### Ethics Policy

#### **Grade marking**

The University uses an alpha numeric grade scale for the marking of assessments. Unless you have been specifically informed otherwise your marked assignment will be awarded a letter/number grade. More detailed information on grade marking and the grade scale can be found on the portal and in the Student Handbook.

#### Grade Marking Scale

#### **Guidance for online submission through Solent Online Learning (SOL)**

#### Online Submission