ROBERT GORDON UNIVERSITY ABERDEEN

Assessment Brief - Coursework

Academic Year	**2024-25**
Semester	**2**
Module Number	**CM1601**
Module Title	**Programming Fundamentals**
Assessment Method	**Individual Coursework comprising of two working prototypes and a report.**
Deadline (time and date)	**Monday, 28th April 2025** **1.00 pm**
Submission	**Assessment Dropbox in the Module Study Area in Moodle.**
Word Limit	**Page Limit is twenty pages**
Use of Generative Artificial Intelligence (AI) text	**Authorised**.
Module Co-ordinator	**Cassim Farook**

What knowledge and/or skills will I develop by undertaking the assessment?

Describe the knowledge and/or skills that students will develop by undertaking the assessment.

- Develop knowledge of **design** using Flow Charts, Pseudocode and Class Diagrams for the proposed solution.
- Gain skills of using **python** language to construct a basic menu driven command line application.
- Gain skills of using use **java** language to construct a basic menu driven command line application and using a **java FX** to enhance a chosen functionality with a graphical user interface.
- Gain skills of developing "interoperable" systems using a shared **text file for data exchange**.
- Improve source code version management skill by using **Git with GitHub**.
- Developing **Test-Cases** and then translating them to **unit-tests** to test the developed solution.
- Improve documentation and presenting skills by producing a technical document and presenting the solution in form of a demo to an audience.

On successful completion of the assessment students will be able to achieve the following Learning Outcomes:

What knowledge and/or skills will I develop by undertaking the assessment?

2. Apply fundamental programming concepts using a high-level programming language.

3. Implement robust, maintainable programs that use object-orientated analysis and design principles.

Please also refer to the Module Descriptor, available from the module Moodle study area.

What is expected of me in this assessment?

Task(s) - content

The Sri Lanka Budget 2025 states that the use of Point-of-Sale (POS) machines across businesses, especially in VAT-registered enterprises, will be implemented as a key initiative to facilitate digital transactions and reduce cash dependency. In addition, the Budget also states that steps will be taken toward digitalizing tax systems of the government to reduce leakages and enhance transparency while minimizing human interactions in tax administration (Newswire, 2025).
A self-employed cake maker who sells cupcakes for a living is now forced to procure a simple point-of-sale system to manage transactions relate to sales. You are entrusted with the task in developing the software that will have the additional capability of generating a "tax text file" to be sent to the tax department. The POS system is to be a simple command line program to be developed using Python programming language.
To ensure inter-operability, you are also required to implement the tax file processing system module as part of the government tax department digitizing initiative. The tax department only authorizes software that is developed using Java and they mandate it must use WIMP - windows, icons, menus, and pointer. They have pre-approved using Java FX based WIMP programs.

Requirements for the Python **command line** POS system.

a) **Add** To Basket (item code, internal price, discount, sale price, quantity). Item code consists of characters (uppercase and/or lowercase) and with/without numbers, underscores.
   Example of item codes are: Lemon_01 , LE_cup01, Cake124
   i. System should calculate the line total as sale price multiplied by quantity.
   ii. System should allow to add multiple items.
   iii. The system should show currently added items with a line number.
b) **Delete** Item from Basket – When the line number is given, the line item would be removed.
c) **Update** Item – When the line number is given, the sale price, discount and quantity can be updated.
d) Generate Bill – The items that are added to the basked are finalized and the grand total is calculated. The Bill number is generated and shown to the user. The line items will be saved with the bill number, and the basket will be emptied.
e) **Search** Bill – The system will show the associated line items when the correct bill number is given.
f) Generate Tax Transaction File – This will generate a text file with a checksum for each record.

When generating the tax transaction file a "checksum" is appended to each line. A simple **checksum algorithm** is given below Wikipedia (2020).

What is expected of me in this assessment?

Checksum algorithm

For a transaction line:
CALCULATE the count of all capital letters
CALCULATE the count of all simple letters
CALCULATE the count of all numbers and decimals
SUM the above three values
This will give the checksum for the transaction line.

The checksum is appended to each transaction and the a ASCII file is created. The data in the file can be in **one of the following structured format**.

a) CSV – put the checksum at the end of the line for each record.
b) TSV – put the checksum at the end of the line for each record.
c) JSON – put the checksum in a attribute called "checksum" in the JSON transaction string.
d) XML – put the checksum in a element called "checksum" in the XML transaction string.

Requirements for the JavaFX Government Tax Department System (**JFXGTDS**) Application.

g) A function to "import" the transaction file. You can give the full path to the file in a textbox.
h) A screen that will show the imported transactions (item code, internal price, discount, sale price, quantity, checksum).
i) A function to validate if each transaction record is accurate. The screen should show a summary of:
    i. Total records that were imported.
    ii. Total Valid records.
    iii. Total Invalid records.

    Invalid records rules: (you can manually edit the text file before importing trigger these rules)
    - For each row you should calculate the checksum, compare with the original checksum value for that row. If checksum does not match the transaction line is treated as invalid.
    - The transactions that contain a special character in the item code.
    - The transactions that contain a minus value for the item price.

j) The manager should be able to **update** invalid records. When editing the manager should be able to edit all fields and save. The checksum should get calculated again and updated for that row.
k) The manager should be able to **delete** invalid records.
l) A function to calculate the profit value for each transaction. The calculated profit need to be stored for each transaction. Profit = (internal price * quantity) – (sale price * quantity – discount). You would need to make sure you have such transactions. Profit can be positive or negative (negative profit is treated as a "Loss".
m) A function to delete transactions where the profit is zero.
n) A function to calculate the final tax. The tax rate is a input by the manager.
    Final Tax = [For all line items SUM of (Profit – Loss)] * tax rate%.

Reference

Newswire (2025). *Budget 2025 : Live updates - Newswire*. [online] Available at: https://www.newswire.lk/2025/02/17/budget-2025-live-updates/.
Reuters (2025). Sri Lanka scrambles to restore power after monkey causes island wide outage. *Reuters*. [online] 13 Feb. Available at: https://www.reuters.com/world/asia-pacific/sri-lanka-scrambles-restore-power-after-monkey-causes-islandwide-outage-2025-02-13/.
Wikipedia (2020). *Checksum*. [online] Available at: https://en.wikipedia.org/wiki/Checksum.

What is expected of me in this assessment?

Task(s) – format

1. Developed Program in a Zip File

1.1. Pycharm to be used for Python development. IntelliJ to be used for JavaFX development.

1.2. Private GitHub repository to be used for version controlling the source code. You must give access to:

- kalhari.w@iit.ac.lk
- prashastha.m@iit.ac.lk
- prasan.y@iit.ac.lk
- akarshani.r@iit.ac.lk

1.3. Unit-Tests with JUnit to be written to test the Java application.

1.4. You should NOT use a database - it is assessed in another module.

1.5. It is required to write classes with functions in both applications. You can use supporting libraires however you must justify when questioned.

1.6. To gain higher grades, you need to use all the control constructs, looping constructs, branching constructs, validation, and exception handling in both applications.

2. PDF Report

A report will consist of the following.

2.1. The “private” Git Hub URL and Screenshots of the Git Hub repository where you maintained the source code.

2.2. If you did not use a Generative AI (GA) software, indicate it in the report. If GA was used, indicate the URL of the product or the software name and version you used. You must record the prompts that you used on it. To the report you only put the prompts that was very helpful to you.

2.3. One Flow chart for POS system, from adding an item to producing the Tax Transaction File.

2.4. One Flow chart for Government Tax system, from importing the Tax Transaction file to calculating the Final Tax.

2.5. One Class diagram for the Government Tax Department system.

2.6. One test case you wrote to desk check a function of the POS system. You may choose any function.

2.7. One test case you wrote for desk checking the Government Tax system. You may choose any function.

2.8. Three selected Screenshots of the POS system application in action.

2.9. Three selected Screenshots of the Government Tax system in action.

2.10. One screenshot of the data contained in the Tax Transaction file.

How will I be graded?	
The overall grade for the assessment will be calculated using the algorithm below.	
A	At least 50% of the subgrades to be at Grade A, at least 75% of the subgrades to be at Grade B or better, and normally 100% of the subgrades to be at Grade C or better.
B	At least 50% of the subgrades to be at Grade B or better, at least 75% of the subgrades to be at Grade C or better, and normally 100% of the subgrades to be at Grade D or better.
C	At least 50% of the subgrades to be at Grade C or better, and at least 75% of the subgrades to be at Grade D or better.
D	At least 50% of the subgrades to be at Grade D or better, and at least 75% of the subgrades to be at Grade E or better.
E	At least 50% of the subgrades to be at Grade E or better.
F	Failing to achieve at least 50% of the subgrades to be at Grade E or better.
NS	Non-submission.

*If the word count is above the specified word limit by more than 10% or the submission contains an excessive use of text within tables, the grade for the submission will be reduced to the next lowest grade.

ROBERT GORDON UNIVERSITY ABERDEEN

Feedback grid / Marking Scheme

GRADE	A	B	C	D	E	F
DEFINITION / CRITERIA (WEIGHTING)	**EXCELLENT** Outstanding Performance	**COMMENDABLE/VERY GOOD** Meritorious Performance	**GOOD** Highly Competent Performance	**SATISFACTORY** Competent Performance	**BORDERLINE FAIL**	**UNSATISFACTORY** Fail
Flow Chart for POS and JFXGTDS (2 subgrade)	Fully detailed, complete and excellent flow chart covering ALL functionality requested, inclusive checksum algorithm, file export functionality for the POS system. Complete and excellent flow chart covering ALL functionality, inclusive of file import, checksum verification and final tax calculation algorithms for the JFXGTDS. All control constructs, conditional statements, branch constructs are evidenced. Relevant additional functionality are designed using flowcharts to support the implementation. Accurate use of symbols.	Fully detailed, complete and excellent flow chart covering ALL functionality requested, inclusive checksum algorithm, file export functionality for the POS system. Complete and excellent flow chart covering ALL functionality, inclusive of file import, checksum verification and final tax calculation algorithms for the JFXGTDS. Partial usage of control constructs, conditional statements, branch constructs evidenced. Accurate use of symbols. The flow charts are detailed with variables and condition notes to enable figuring out a corresponding pseudocode just by looking at it.	A detailed flow chart for POS system, from adding an item to producing the Tax Transaction File is documented. An detailed flow chart for JFXGTDS,, from importing the Tax Transaction file to calculating the Final Tax is documented. Some justification is provided why a certain or validation loop/routine flow was added within the main flows.	An abstract level flow chart for POS system, from adding an item to producing the Tax Transaction File is documented. An abstract level flow chart for JFXGTDS from importing the Tax Transaction file to calculating the Final Tax is documented. There may be some inconsistencies or drawbacks on the notations.	At least one flow chart exists for either POS or JFXGTDS	Some or no attempt at a flow chart
Implementation of the POS (2 subgrades)	Complete and excellent Implementation of the POS. Excellent flow and usability. Menu driven. Version control is used effectively.	Complete and very good Implementation. Very good user interface. There are no Syntax errors. Runtime errors or Semantic errors	Complete implementation for best case scenario where valid inputs produces correct results.	Implementation is evident with some minor logic errors.	Code exists for the POS system but does not work.	Code does not exists or exists in isolation.
Tax Transaction File Export (1 subgrade)	Complete and excellent implementation of Tax Transaction File generation. Advanced checksum algorithm has been designed and implemented. Manual tampering is detected as invalid when importing.	Very good implementation of the export function inclusive of the given checksum algorithm. Special characters that will cause issues with the file format is handled on export/import.	Complete and good implementation of the export function with the given checksum algorithm. Import works.	Export function exists. Import works. Some errors exists.	Incomplete implementation of the file.	Incomplete and unsatisfactory of implementation of the Tax Transaction File

GRADE	A	B	C	D	E	F
DEFINITION / CRITERIA (WEIGHTING)	**EXCELLENT** Outstanding Performance	**COMMENDABLE/VERY GOOD** Meritorious Performance	**GOOD** Highly Competent Performance	**SATISFACTORY** Competent Performance	**BORDERLINE FAIL**	**UNSATISFACTORY** Fail
Implementation of the JFXGTDS (2 subgrades)	Complete and excellent implementation of JFXGTDS. Excellent flow and usability. Menu driven. Version control is used effectively.	Complete and very good Implementation. Very good user interface. There are no Syntax errors. Runtime errors or Semantic errors	Complete implementation for best case scenario where valid inputs produces correct results.	Implementation is evident with some minor logic errors.	Code exists for the JFXGTDS system but does not work.	Code does not exists or exists in isolation.
Class Diagram For JFXGTDS (1 subgrades)	Complete and Detailed Class diagram for the JFXGTDS. Correct identification of classes, attributes, its data types, method signatures and scope (public, private) identified. Class diagram corresponds to the actual implementation.	Very good Class diagram for the JFXGTDS. Correct identification of classes, attributes, its data types, method signatures. There is some resemblance to the classes implemented in the JFXGTDS	Class diagram exists with good use of the notations. Attributes and method signatures identified. Some justifications given.	Class diagram exists. At least three classes identified. Adequate use of notations.	Class diagram exists, but totally incorrect notations.	Not found.
Testing (1 subgrades)	Complete and excellent documentation and execution of the Desk check process. Java Unit tests are conducted and evidenced.	Very good documentation and execution of the Desk check process. Java Unit tests are conducted and evidenced.	Good documentation and execution of the Desk check process. Some form of test cases found.	Some sort of testing is conducted and evidenced in the report.	Testing is evidenced but not conducted.	No evidence of testing.
Presentation and Q&A (1 subgrade) **Overriding Grade**	Presented with very good explanations with justifications and confidence. Excellent understanding of programming principals.	Presented well with less justifications and explanations. Very good understanding of programming principals.	Presented moderately well the adequate explanations Good understanding of programming principals.	demonstrated adequate understanding of programming principals	Some attempt was made to explain.	Clueless

If the student was absent for the live viva demonstration, the entire module grade will be marked as "NS"

Overriding Grade : The grade given for the Overriding Mark supersedes the calculated grade based on the students performance at the defence. This is to ensure that the student, regardless of using Generative AI or doing on his own showcases majority value addition and thorough understanding of programming principles.

Coursework received late, will be regarded as a non-submission (NS) and one of your assessment opportunities will be lost. You must use the coursework extension path for submitting late. Contact SRU or your level coordinator for the correct procedure.

What else is important to my assessment?

What is the Assessment Word Limit Statement?

It is important that you adhere to the Word Limit specified above. The Assessment Word Limit Statement can be found in Appendix 2 of the RGU Assessment Policy. It provides detail on the purpose, setting and implementation of wordage limits; lists what is included and excluded from the word count; and the penalty for exceeding the word count.

What's included in the word count?

The table below lists the constituent parts which are included and excluded from the word limit of a Coursework; more detail can be found in the full Assessment Word Limit Statement. Images will not be allowed as a mechanism to circumvent the word count.

Excluded	Included
Cover or Title Page	Main Text e.g. Introduction, Literature Review, Methodology, Results, Discussion, Analysis, Conclusions, and Recommendations
Executive Summary (Reports) or Abstract	Headings and subheadings
Contents Page	In-text citations
List of Abbreviations and/or List of Acronyms	Footnotes (relating to in-text footnote numbers)
List of Tables and/or List of Figures	Quotes and quotations written within "…"
Tables – numeric content	Tables – text content
Figures	
Reference List and/or Bibliography	
Appendices	
Glossary	

What are the penalties?

The grade for the submission will be reduced to the next lowest grade if:

- The word count of submitted work is above the specified word limit by more than 10%.
- The submission contains an excessive use of text within Tables or Footnotes.

What else is important to my assessment?

What is plagiarism?

Plagiarism is "the practice of presenting the thoughts, writings or other output of another or others as original, without acknowledgement of their source(s) at the point of their use in the student's work. All materials including text, data, diagrams or other illustrations used to support a piece of work, whether from a printed publication or from electronic media, should be appropriately identified and referenced and should not normally be copied directly unless as an acknowledged quotation. Text, opinions or ideas translated into the words of the individual student should in all cases acknowledge the original source" (RGU 2022).

What is collusion?

"Collusion is defined as two or more people working together with the intention of deceiving another. Within the academic environment this can occur when students work with others on an assignment, or part of an assignment, that is intended to be completed separately" (RGU 2022).

For further information please see Academic Integrity.

What if I'm unable to submit?

- The University operates a Fit to Sit Policy which means that if you undertake an assessment then you are declaring yourself well enough to do so.
- If you require an extension, you should complete and submit a Coursework Extension Form. This form is available on the RGU Student and Applicant Forms page.
- Further support is available from your Course Leader.

What additional support is available?

- RGU Study Skills provide advice and guidance on academic writing, study skills, maths and statistics and basic IT.
- RGU Library guidance on referencing and citing.
- The Inclusion Centre: Disability & Dyslexia.
- Your Module Coordinator, Course Leader and designated Personal Tutor can also provide support.

What are the University rules on assessment?

The University Regulation 'A4: Assessment and Recommendations of Assessment Boards' sets out important information about assessment and how it is conducted across the University.