ROBERT GORDON UNIVERSITY ABERDEEN

Assessment Brief Template

Academic Year	**2023**
Semester	**2**
Module Number	**CM1601**
Module Title	**Programming Fundamentals**
Assessment Method	**Course Work**
Deadline (time and date)	**5th April 2023**
Submission	**Assessment Dropbox in the Module Study Area in Campus Moodle.**
Word Limit (see Assessment Word Limit Statement)	**3000 words**
Module Co-ordinator	**Iresh Bandara**

What knowledge and/or skills will I develop by undertaking the assessment?
1. Demonstrate competence in the algorithmic approach to problem solving. 2. Design, code, compile, test and execute fundamental programming concepts using a high-level programming language. 3. Construct robust, maintainable programs that use object-orientated analysis and design principles.
On successful completion of the assessment students will be able to achieve the following Learning Outcomes: 1. Design, code, compile, test and execute fundamental programming concepts using a high-level programming language. 2. Construct 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

Please note that this is a continuation of Coursework 1. You are instructed to use Java and OOP concepts to create a GUI application for the World rally cross championship management.

Objective

You are instructed to create a GUI application for a World rally cross championship management using JAVA along with a detailed report.

Features of the system
System should have a JAVA FX GUI for the following functions when launched.

Adding driver details
Deleting driver details
Updating driver details
Viewing the rally cross standings table
Simulating a random race
Viewing race table sorted according to the date
Save the current data to a text file
Load data from the saved text file
Exit the program.

Coursework Sections	Detailed content	Learning Outcome	Topic/Week Module Activity
Adding driver details	System should allow the user to enter the following details (not limiting to) by prompting the required information. Name, age, team, car, current points *example: Travis Pastrana, 38, Subaru Motorsports, Subaru WRX STi, 10*	1	Lecture 2 – Operators and Expressions Lecture 7 - Methods
Deleting driver details	System should allow the user to delete a driver by searching by name.	1	Lecture 2 - Operators and Expressions
Updating driver details	System should allow the user to update driver details by searching by name.	1	Lecture 2 - Operators and Expressions
Viewing the rally cross standings table	System should display the championship standings ordered by points in descending order. This table should be formatted neatly with all the details.	1	Lecture 7 : Java Strings
Simulating a random race	System should simulate a random race and assign pints to each driver accordingly. First place gets 10 points, second place	1	Lecture 2 - Operators and Expressions

What is expected of me in this assessment?

	gets 7 points, third place gets 5 points. A position should be assigned to all the drivers in the drivers list. Details about the races should be stored in a text file and should be loaded automatically when the program starts. A race should have the following details. Date of the race Location of the race (Nyirád, Höljes, Montalegre, Barcelona, Rīga, Norway) Each driver's position and points		Lecture 4 – Control Flow Statements Lecture 5 – Control Flow Statements II Lecture 6- Exception Handling
Viewing race table sorted according to the date	System should display all the races in the championship sorted according to the date. Use your own algorithm for sorting.	1	Lecture 8- Collections
Save the current data to a text file	System should be able to save the current data to a text file in a way that the data can be retrieved easily. Note that you are not allowed to use a database for storing data.	1,2	Lecture 10 – File handling
Load data from the saved text file	System should be able to load the current data from the text file to enable resume capabilities.	1,2	Lecture 10 – File handling

Task(s) - format

1. All the functionalities mentioned above need to be fully implemented. Code should be submitted as a **single zip file** to the link in assessment section. Check the following key points before you start the implementation

    a. A GUI system to enable the user input with validations should be implemented.
    b. Information entered should be saved in text file(s).
    c. You are not allowed to use databases for this coursework.
    d. When sorting data, you are not allowed to use inbuilt Java libraries and marks will be given only for your own algorithms.
    e. All the functionalities which were developed earlier in your last coursework need to be implemented again.
    f. You are allowed to use Java FX for implementing the user interfaces. You have the freedom to design the interfaces.
    g. When implementing the application, you are instructed to use classes and objects and data structures with OOP concepts for
        a. Classes and objects: Mandatory to use
        b. Encapsulation: Mandatory to use
        c. Inheritance: Optional
        d. Abstraction and polymorphism: Optional
    h. Create a Junit test suit to cover the whole scenario. Include in the report.

What is expected of me in this assessment?

2. You should submit a report containing the following.

    a. Write a test plan to cover the whole scenario. Each test case should have a name, inputs, expected output and the actual output.
    b. Your complete java code for each function with a brief description.
    c. Explain what you have done in order to ensure the robustness and the maintainability of your source code. Mainly explain whether you followed the guidelines you were taught during the code quality lecture sessions. Further you have followed SOLID principles when designing the system or not. Feel free to refer online resources when justifying.

3. Coursework Report Format:
    a. Your report should be word-processed, font size 11, 1.5 line spaced, referenced and uploaded as ONE final document to the Assessment drop box in the Assessment area of the Moodle page as PDF file.

4. Report Structure
    a. Title Page
    b. Executive Summary
    c. Contents Page
    d. Flow charts
    e. Introduction to functions with code
    f. J-units, Test plan and test cases
    g. Robustness and the maintainability
    h. Conclusions & assumptions
    i. Reference list
    j. Appendices (if appropriate – 6 pages maximum)

Important: If you are further developing the application or doing any modifications due to a justifiable fact, please mention that in your report too.

The approaches taken, assumptions made, will be checked thoroughly during the viva. Full marks will be awarded for each criterion only if you implement successfully and defend well during the viva.

Please refer to the Word Limit Statement for guidance on what is included and excluded from the word count.

References
Use RGU Harvard style to format citations and references – see library guide (http://libguides.rgu.ac.uk/rguharvard). Refer to textbooks and articles/academic papers rather than lecture notes.

How will I be graded?	
A grade will be provided for each criterion on the feedback grid which is specific to the assessment. The overall grade for the assessment will be calculated using the algorithm below. *[Amend as appropriate to your module.]*	
A	At least 50% of the feedback grid to be at Grade A, at least 75% of the feedback grid to be at Grade B or better, and normally 100% of the feedback grid to be at Grade C or better.
B	At least 50% of the feedback grid to be at Grade B or better, at least 75% of the feedback grid to be at Grade C or better, and normally 100% of the feedback grid to be at Grade D or better.
C	At least 50% of the feedback grid to be at Grade C or better, and at least 75% of the feedback grid to be at Grade D or better.
D	At least 50% of the feedback grid to be at Grade D or better, and at least 75% of the feedback grid to be at Grade E or better.
E	At least 50% of the feedback grid to be at Grade E or better.
F	Failing to achieve at least 50% of the feedback grid to be at Grade E or better.
NS	Non-submission.

ROBERT GORDON UNIVERSITY ABERDEEN

Feedback grid

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
A GUI menu system with validations **(15 x %)** **Grade:** ☐	All the Options are displayed on a Java FX GUI neatly and correctly. Users can select the option by clicking, navigating to a page or tab. Inputs are validated. The student demonstrates excellent knowledge on the task trough viva. Beautifully crafted GUIs.	More than 7 Options are displayed on a JAVA FX GUI correctly. Users can select the option by clicking, navigating to a page or tab. basic input validation can be seen. The student demonstrates very good knowledge on the task through viva. GUIs are designed nicely.	More than 6 Options are displayed on a JAVA FX GUI correctly. Users can select the option by clicking, navigating to a page or tab. basic input validation. The student demonstrates good knowledge on the task trough viva. Good design of GUIs.	More than 5 Options are displayed on a JAVA FX GUI correctly. Users can select the option by clicking. minor input validation. The student demonstrates adequate knowledge on the task through viva. Satisfactory GUIs	Less than 5 Options are displayed on a Console menu. Users can only select a few options via input. no input validations can be seen. The student does not demonstrate adequate knowledge on the task through viva.	No GUI menu to be seen. user inputs are not considered. Demonstrates no knowledge on the task. Program does not run.
Add, delete & update driver details **(10x %)** **Grade:** ☐	Adding, deleting, and updating of driver details can be done by searching. Duplicate driver records are handled, and all the inputs are correctly validated. sub-menu for each function works flawlessly. The student demonstrates excellent knowledge on the task trough viva.	Adding, deleting, and updating of driver details can be done by a single attribute. duplicate driver records are handled, and all the inputs are correctly validated. sub-menu for each function works. The student demonstrates very good knowledge on the task trough viva	Adding, deleting, and updating of driver details can be done. No duplicate driver records are handled. Basic inputs are validated. sub-menu for each function works. The student demonstrates good knowledge on the task trough viva	Adding, deleting, and updating of driver details can be done. No duplicate driver records are handled. no inputs are validated. sub-menu for each function works. The student demonstrates adequate knowledge on the task trough viva	only two of the functions out of three can be done. No duplicate driver records are handled. Basic inputs are not validated. sub-menu for each function works with errors. The student demonstrates low knowledge on the task trough viva	Functions do not work. Demonstrates no knowledge on the task. Program does not run.

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
Standing table and race table **(10x %)** **Grade:** ☐	Standing and race tables are neatly formatted to imitate a tabular format. Data are sorted according to the Points and date respectively. Student has implemented two sorting algorithms for the standings and race table. All the required columns can be seen in the tables. The student demonstrates excellent knowledge on the task trough viva.	Standing and race tables are neatly formatted. Data are sorted according to the Points and date respectively. Student has implemented at least one sorting algorithms for the standings or race table. required columns can be seen in the tables. The student demonstrates very good knowledge on the task trough viva.	Standing and race tables are printed on to the console. Data are sorted according to the Points and date respectively. Student has implemented at least one sorting algorithms for the standings or race table. Most of the required columns can be seen in the output. The student demonstrates good knowledge on the task trough viva.	Standing and race tables are printed on to the console. Data are sorted according to the Points and date respectively but by using the default sort function. Most of the required columns can be seen in the output. The student demonstrates satisfactory knowledge on the task trough viva.	Standing or race table is printed on to the console with nor formatting. No sorting can be seen, or sorting is incorrect. Most of the required columns are missing in the output. The student demonstrates low knowledge on the task trough viva.	Standing or race table is not printed on to the console. student demonstrates no knowledge on the task trough viva.
Random Race (15x %) **Grade:** ☐	Random race is generated with all the players in the system with winning positions. Points are updated after each race. Duplicate date has been validated. Race details are stored in the file correctly. Correct locations are randomly selected. Student has developed an algorithm for this function. All the validations are done. The student demonstrates excellent	Random race is generated with all the players in the system with winning positions. Points are updated after each race. Duplicate date has been validated. Correct locations are randomly selected. The student demonstrates very good knowledge on the task trough viva.	Random race is generated with all the players in the system with positions. Points are updated after each race. Locations are randomly selected. The student demonstrates good knowledge on the task trough viva.	Random race is generated with most of the players in the system with positions. No validations can be seen. Points are updated after the race is generated. Locations are randomly selected. The student demonstrates satisfactory knowledge on the task trough viva.	Random race is generated with no relevance to the existing data. Positions are incorrectly calculated. No validations can be seen. Points are not updated after the race. The student demonstrates unsatisfactory knowledge on the task trough viva.	Random race function is not implemented. student demonstrates no knowledge on the task trough viva.

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
	knowledge on the task trough viva.					
File handling (10x %) **Grade** ☐	Serializable text files have been used to store the relevant data. Necessary validations have been implemented and possible duplicate errors were addressed. Exceptions were handled in the code. Resume capability of the system have been implemented. The student demonstrates excellent knowledge on the task trough viva.	Clear text or xml files have been used to store the relevant data. Necessary validations have been implemented and possible duplicate errors were addressed. Resume capability of the system have been implemented. The student demonstrates very good knowledge on the task trough viva.	Clear text files have been used to store data. validations were implemented. Resume capability of the system have been implemented. The student demonstrates good knowledge on the task trough viva.	At least one text file has been used to store data. Basic validations can be seen. Resume capability of the system have been implemented to a functioning level. The student demonstrates satisfactory knowledge on the task trough viva.	Attempts were made to code the file handling but not working. File has been created without data. The student demonstrates unsatisfactory knowledge on the task trough viva.	File handling is not implemented. student demonstrates no knowledge on the task trough viva.
Report & Test Plan (15x %) **Grade** ☐	A Detailed, fully covered test plan can be seen in the report. Excellent justification is given in the report on selecting test cases. Report has all the required details with references. Code is added to the report and clear detailed descriptions are given. Excellent formatting can be seen throughout the report. J-Units are correct and covered.	A test plan covering most of the functions can be seen in the report. good justification is given in the report on selecting test cases. Report has the required details with references. Code is added to the report with clear descriptions. no formatting errors can be seen. Both Flow charts are correct. J-Units are correct and covered.	A test plan covering most of the functions can be seen in the report. Basic justification is given in the report on selecting test cases. Report has the required details. Code is added to the report with brief descriptions. minor formatting errors can be seen. J-Units are correct, and majority of the functions are covered.	A test plan covering main functions can be seen in the report. Some justification is given in the report on selecting test cases. Report has the basic details. Code is added to the report with short descriptions. Minor formatting errors can be seen. J-Units are correct, and half of the functions are covered.	Incomplete report. Test plan does not cover even the main functions. Inadequate justifications are given. No references given. only few code fragments attached. Major formatting errors can be seen throughout the document. No J-Units	No test plan to be found. No justification given. No code is attached, or code attached as screen shots. No J-units

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
OOP Usage (15x %) **Grade** ☐	Abstraction, Inheritance, Polymorphism and Encapsulation has been used throughout the CW implementation. Interfaces has been used. The student demonstrates excellent knowledge on OOP trough viva.	Al least 3 OOP concepts has been used. At least one interface has been used. The student demonstrates very good knowledge on OOP trough viva.	Abstraction and Encapsulation has been used throughout the CW implementation. At least one interface has been used. The student demonstrates good knowledge on OOP trough viva	Abstraction or Encapsulation has been used in the CW implementation. The student demonstrates satisfactory knowledge on OOP trough viva	No visible OOP concept usage throughout the usage. Even if used, purpose has been misled. The student demonstrates unsatisfactory knowledge on OOP trough viva.	No OOP Usage. student demonstrates no knowledge on the OOP trough viva.
Code Quality and improvements (10x %) **Grade** ☐	Code comments, in code references, Proper naming conventions, Indentation and use of proper coding structures can be seen. Smart Improvements on the given specification is evident.	Code comments, Proper naming conventions, Indentation can be seen. Many Improvements on the given specification is evident.	Code comments, Proper naming conventions, Indentation can be seen. At least one Improvements on the given specification is evident.	Code comments, Proper naming conventions, Indentation can be seen.	No comments to be seen. indentation is not consistent. Major naming errors are visible.	No comments to be seen. No indentation of the code. has no idea on naming conventions.

Coursework received late, without valid reason, will be regarded as a non-submission (NS) and one of your assessment opportunities will be lost.

ROBERT GORDON UNIVERSITY ABERDEEN

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 is the Assessment Word Limit Statement?

It is important that you adhere to the Word Limit specified above. The Assessment Word Limit Statement lists what is included and excluded from the word count, along with the penalty for exceeding the upper limit.

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 else is important to my assessment?

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.