Computer Programming
Exam Project Evaluation
Summer 2019

Criteria for grading

14 points in total - Minimum to pass 8, according to 5 dimensions:

1) Functionality (Coverage of the requirements)

A: all functions implemented
B: (-1 point) if a minor functional specification or requirement is not implemented or not correctly implemented (logical errors)
C: (-2 points) if 2 functions or requirements are not implemented or not correctly implemented
D: (-3 points) if 3 functions or requirements are not implemented or not correctly implemented
E: (-4 points) if 4 functions or requirements are not implemented or not correctly implemented
F: if many functions or requirements are not implemented and the project is very much smaller than the requested one

2) GUI usability

A: if the GUI is well designed
B: (-1 point) if there are some small issues, such as labels that cannot be read easily or strange names.
C: (-2 points) When there are small issues and it is not immediate to understand how a function can be executed through the GUI.
D: (-3 points) If there are more severe problems, such as windows that are popping up without content of labels that are not shown
Fail: severe problems such as the GUI blocks or you cannot go back to the beginning of the interaction state

3) Errors and exceptions

A: if no run time errors or exceptions
B: (-1 point) If you got one or more run time error or exception but the application is still running.
C: (-2 points) If many errors or exceptions are raised.
Fail: There are many exceptions and the application cannot be used.

4) Code structure

A if the code is well written and well structured (e.g., a reasonable number of classes is used and the naming conventions are applied)
B (-1 point) if the student has not created a reasonable set of classes or the code is messy with strange names and unclear method calls

5) Fix to Start:

Fail: If the application does not start or some necessary files are not found.

In addition to the points from the project, the students may get up to 3 points from the lab assignments (explained in the lab 1) up to 14 points.

Formula for the project evaluation:
Project mark = 14 - negative points
Final mark = min(Project mark + lab points, 14)

Example:
A student gets the following scores:
Functionality: B (-1) Standard deviation was not calculated.
User Interface: B (-1) The GUI for entering the replies is messy.
Exception: C (-2) many errors or exceptions are raised.
Quality of the Code: A
14 - 4 = 10
Final mark: 10 + 1 from lab assignment = 11

------------------------

Student: 16849
Functionality: D There is no main window menu; the 'about' functionality is implemented with a button (but the number of methods and lines of code is not displayed). The number of decimal places can be chosen, but there is no effect on the output. The transpose is not working if the matrix is not square and also it does not work if the other matrix area is empty. Multiplication does not work correctly when matrices are not square.
User Interface: B The result of the matrices operation is shown in a dialogue; this is not a good choice (we suggested a different implementation).
Exception: B One exception was raised when I tried to compute the transpose of a matrix.
Quality of the Code: A The code is reasonably written.
14 - 5 = 9
Final mark: 9 + 2.4 from lab assignment = 11

------------------------

Student: 17113
Functionality: E There is no main window menu; the 'about' and 'exit' functionality is implemented with a button. It is not possible to select the number of decimal digits for the output. The functionality of showing the log of the performed operations is not implemented. The result of the operation is not shown on a separate area (as requested) but it replaces the first matrix.
User Interface: A
Exception: No exceptions
Quality of the Code: A code is OK
14 - 5 = 9
Final mark: 9 + 2.4 from lab assignment = 11

------------------------
14 - 4 = 10

Final mark: 10 + 0.6 from lab assignment = 11

------------------------