cct | College Dublin
Computing • IT • Business

CONTINUOUS ASSESSMENT / ASSIGNMENT

Programme Title/Year:	**BSc. in IT – 1st Year**
Module Title(s):	**Computer Programming** **Problem Solving & Mathematics for IT**
Lecturer Name(s)	**Ken Healy** **Carole McLoughlin**
Assessment Title:	**Matrix Calculator**
Assessment Type:	**INDIVIDUAL**
Assessment Weighting:	**Computer Programming – 25%** **Problem Solving & Mathematics for IT – 25%**
Date Issued:	**4th February 2019**
Due Date (Deadline):	**Midnight on 16th March 2019**
Late Submission Penalty:	Late submissions will be accepted up to **5 days** after the deadline. All late submissions are subject to a penalty of **10%** per day. Submissions received more than 5 days after the deadline above **will not** be accepted.
Method of Submission:	**Moodle**
Feedback Method:	**Results posted in Moodle gradebook**
Instructions for Submission:	**Computer Programming:** **Create a folder using your STUDENT ID as the folder name.** **Copy the relevant .java file to your folder. All code and comments must be included in the .java file. Other file types WILL NOT BE MARKED.** **ZIP (or compress) the folder and submit the ZIPPED folder on Moodle for Computer Programming.**

Module Learning Outcomes Assessed:

Computer Programming:

- Implement the core syntax and semantics of the programming language utilised
- Debug programs in a meaningful way using known methods and tools, diagnose and resolve syntax, logical and runtime errors
- Approach and solve problems in a structured way under given constraints
- Build robust software that adheres to current conventions, that is reliable, maintainable, available and if necessary expandable and reusable
- Identify and use good principles of algorithm design to write and debug well-structured programs

NOTE: The requirements for Problem Solving & Mathematics for IT are posted separately in that module. These are linked to this assignment but the requirements are separate and you must make a separate submission for that module.

Assignment Detail – COMPUTER PROGRAMMING

The objective of this assignment is to create a program that will work as a Matrix Calculator

INPUT

1) The Matrix Calculator MUST take its input from a File. This file will follow a pre-defined format (see appendix)

   MINIMUM REQUIREMENT: The program will read in Matrix data from a file named "matrix.txt" and store the data within the program so that calculations can be carried out.

   IMPORTANT: **The import must occur AFTER the user has selected a calculation option (see number 2 below).**

   DISTINCTION WORK (OPTIONAL): The user will be asked to provide a filename and the program will read the data from this file and store it within the program so that calculations can be carried out.

2) The user will be prompted (on screen) to choose from a list of Calculation options (i.e. display a menu of options).

   MINIMUM REQUIREMENT - The options will be as follows:

   - Add two matrices
   - Subtract Matrices
   - Scalar Multiplication of a Matrix (or Multiply a matrix by an Integer)
   - Multiply two matrices
   - QUIT PROGRAM

   OPTIONAL (Distinction Work):

   The menu will include options for:

   - Scalar Division
   - Calculate the Reciprocal (or Inverse) of a 2 x 2 Matrix

   If the user selects the “QUIT Program” option then the program should display a goodbye message and end.
   In every other case, the program should carry out the chosen operation (see section on PROCESSING). The answer MUST be output to a file (see section on OUTPUT).

   When the calculation is finished, the program should display a message informing the user where to locate the answer AND THEN display the menu options again. The program should ONLY end when the user selects the “Quit” option.

VALIDATION

Data Input Validation - Minimum Requirement: All data input into the program must be validated appropriately to ensure that the program does not crash. In ANY CASE where the data is not valid, the program must display a suitable error message AND prompt the user again for valid data.

Example 1: if the user is prompted to enter a number, then you must check that the input is in fact a number.

Example 2: If the data in the input file is not valid OR does not follow the File Format rules (see appendix), then the user must be prompted to correct the file and try again.

Data Processing Validation – Minimum Requirement: For all the matrix calculations, if there is some reason that the calculation is impossible due to a “maths rule”, then the program must output a message to the screen to tell the user why it was impossible. This message should explain the relevant rule.

PROCESSING

The program must be able to perform the following Matrix calculations

1) **Addition (MINIMUM REQUIREMENT)**

   The program will input TWO matrices (A and B) from the input file.
   The program will check that it is possible to add them together (A + B).
   If yes - The program will output the answer matrix to a file (see section on OUTPUT)
   If no – The program will output a suitable error message to the screen.

2) **Subtraction (MINIMUM REQUIREMENT)**

   The program will input TWO matrices (A and B) from the input file.
   The program will check that it is possible to subtract them (A – B).
   If yes - The program will output the answer matrix to a file (see section on OUTPUT)
   If no – The program will output a suitable error message to the screen.

3) **Scalar Multiplication (Minimum Requirement)**

   The program will input ONE matrix from the input file.
   The program will prompt the user for an integer value
   The program will carry out scalar multiplication on the matrix using the integer input by the user and output the answer matrix to a file (see section on Output)

4) **Scalar Division (Distinction Work)**

   The program will input ONE matrix from the input file.
   The program will prompt the user for an integer value
   The program will carry out scalar division on the matrix using the integer input by the user and output the answer matrix to a file (see section on Output)

5) **Matrix Multiplication (Minimum Requirement)**

   The program will input TWO matrices (A and B) from the input file.
   The program will check that it is possible to multiply them together (A x B).
   If yes - The program will output the answer matrix to a file (see section on OUTPUT)
   If no – The program will output a suitable error message to the screen.

6) **Calculate the Reciprocal (or Inverse) of a 2 x 2 Matrix (Distinction Work)**

   The program will input ONE 2 x 2 matrix from the input file.
   The program will check to see if it is possible to calculate the Inverse.
   If Yes – the program will output the inverse of the matrix to the output file (see OUTPUT section)
   If no – the program will output an appropriate error message to the screen

Maths Rules:

The rules for matrix addition, subtraction, multiplication and Inverse are not provided here. This is information you must have from Maths. You are required to apply this knowledge here. If you do not apply these rules correctly, then you will lose marks.

OUTPUT

- All error message and user prompts must be displayed on screen. Messages must be clear and sensible.
- (Minimum Requirement) Any answer matrix must be output to a file named “answer.txt” and the user must be informed that this has happened.
- The file format must be identical to the format specified for Input (see appendix).
- (DISTINCTION WORK – OPTIONAL) The user is prompted to specify the name of the output file, and any answer matrix is output to the specified file and the user is informed that this has happened.

SPECIFIC RESTRICTIONS – IMPORTANT

1) You are NOT allowed to use any GUI-related code (such as Java swing, JavaFX or similar). If you submit code that uses GUI elements, you will receive a zero mark.

2) Your submission may contain a **MAXIMUM of TWO classes** (i.e. TWO .java files). You are not expected to have two classes, but this is allowed if you wish.

3) **IF** you do decide to have two classes, then one of the classes **MUST be named Matrix.java** and this class MUST define a matrix and contain all the relevant properties and methods of a matrix, including methods to add, subtract, multiply, scalar multiply, scalar divide and find inverse.

4) **It is ESSENTIAL that your code is properly commented. This means that your code is explained clearly to show that you know how it works. If you do not comment your code, you will lose marks and you risk be awarded a zero mark.**

Marking Schedule – PLEASE READ

Description	**Weighting**
Program produces a clear menu system showing all relevant options and returns to the menu after each operation unless use selects “quit” option.	5
*Program is able to correctly **read in** matrix data from file named matrices.txt OR output a message that the file format is incorrect.* *(DISTINCTION – Program will prompt user for a file name and correctly read in matrix data from the specified file, or output a message that file format is incorrect)* **Relevant code is properly commented, AND methods are used appropriately**	3 (5)
Program is able to output any answer matrix to a file named answers.txt using the correct file format.	3

(DISTINCTION – Program will prompt user for a file name and correctly output data to the specified file in the correct format) **Relevant code is properly commented, AND methods are used appropriately**	(5)
Program is able to ADD two matrices together (A + B) if this is possible OR output a message explaining why it is not possible. **Relevant code is properly commented, AND methods are used appropriately**	10
Program is able to SUBTRACT two matrices (A – B) if this is possible OR output a message explaining why it is not possible **Relevant code is properly commented, AND methods are used appropriately**	10
Program is able to MULTIPLY two matrices together (A * B) if this is possible OR output a message explaining why it is not possible **Relevant code is properly commented, AND methods are used appropriately**	15
Program is able to MULTIPLY one matrix by and NUMERIC value provided by the user (i.e. Scalar multiplication) **Relevant code is properly commented, AND methods are used appropriately**	10
Program is able to DIVIDE one matrix by and NUMERIC value provided by the user (i.e. Scalar division) **Relevant code is properly commented, AND methods are used appropriately**	10
Program can calculate the INVERSE of a 2x2 matrix if this is possible OR output a message explaining why it is not possible **Relevant code is properly commented, AND methods are used appropriately**	15
All data input is VALIDATED appropriately, and the user is prompted to provide valid data if the input is not valid. All messages are <u>clear</u> and <u>user friendly</u>.	15
Total	**100%**

Final Note:

This assignment has been issued with a time period of <u>SIX WEEKS</u> for completion. Please do not expect to be able to complete the assignment in a short period of time!

APPENDIX

Matrix file format details

NOTES:

1) You can assume that the input file will be in **.txt** format. You do NOT have to be able to use any other file types.
2) Any output file that you produce should also be in **.txt** format. This means that it should be possible to open the file using Notepad without any corruption or other readability problems.

FILE FORMAT OUTLINE

- The file will contain AT LEAST ONE Matrix in text form.
- The file will contain AT MOST TWO Matrices in text form.
- If there is more than one matrix, there will be an @ symbol, on a separate line, to denote the end of one matrix and that start of the next.
- If there is only one matrix in the file, then there will be NOTHING after the last row of the matrix (i.e. the file will end with the last row of the matrix).
- The first line of a Matrix will contain ONLY TWO integer values. This will specify the number of ROWS and COLUMNS in the matrix (in that order)
- Subsequent lines will contain the Matrix data. Each row will be on a separate line. Each number will be separated by a space.

SAMPLE 1 – Single matrix – 2 X 2

// -- file start -- //

2 2

Two Rows / Two Columns

13 45

Data for row 1, separated by a space

64 -12

Data for row 2, separated by a space

SAMPLE 2 – SINGLE MATRIX – 2 X 3

// -- File Start -- //

2 3

Two Rows / Three Columns

12 32 7

65 -8 123

SAMPLE 3 – TWO MATRICES

// -- File start -- //

2 2

Two Rows / Two Columns

56 12

-2 90

@

@ symbol to denote end of 1st matrix

2 2

Two Rows / Two Columns

12 78

167 -34

SAMPLE 4 – TWO MATRICES

// -- file start -- //

2 1

Two Rows / One Column

23

56

@

@ symbol to denote end of 1st matrix

1 2

78 -12

SAMPLE TEST FILES

A small number of Sample test files are available on Moodle. At least one of these will be used to test your program, HOWEVER please note that I will also use at least one other test file (that you will not see) as part of the testing. The format of all test files will be the same as outlined above.

You are encouraged to generate your own test files and carry out a range of tests to see how well your program works before you submit it.