

Matlab Project Marking Rubric 2021

Code submitted to Matlab Grader is immediately marked for functionality (does it work?) After the due date it will also be marked by a human for style (is it well written?)

You can submit each function up to **6 times**.

If you submit a function more than once, we will mark your **best** solution (in terms of functionality score) that was received prior to the due date. If there are multiple submissions with the same best functionality score, the most recent of them received by the due date will be the one marked for style.

Style (5 marks)

Style marks are awarded based on examining a **random five** functions from the list of functions that you've submitted.

If you submit fewer than five specified functions it isn't possible to earn full marks for style. If you submit five or more functions it is possible to earn full marks for style, even if you have skipped submitting some functions.

Each style category is worth 1 mark, with 0.2 marks awarded per file from the list of your first five functions that satisfies the requirements for that category.

Well written function header comments (1)

There is a well written, easy to understand and complete header comment. To be complete the header comment must do ALL of the following:

- describe the purpose of the function
- describe the input(s) to the function -
describe the output(s) of the function
- include the author's name.

Remember it is acceptable to reuse text from the project specification document as part of your function header. You should check your function header comment by typing `help` followed by your function name at the command line (this will show your header comment).

Comments in the body of the function (1)

There are other comment(s) in the code that help a reader understand the code.

These comments are additional to the header comment. That is there must be at least one other comment (preferably more if the code warrants it). They must also be helpful, i.e.

```
% a comment to earn a mark
```

is not acceptable and would not earn a mark for having a comment that doesn't relate to your code.

Correct indentation (1)

All of the code is indented correctly according to the standard code conventions.

An easy way to ensure correct indentation is to select your code using Ctrl-A and then auto-indent it using the keyboard shortcut Ctrl-I

Appropriate variable names (1)

Appropriate valid variable names have been used to give a good indication of what that variable represents. No specific naming convention is required, eg a variable that contains the total area could be called `total_area`, `TotalArea`, `totalArea`, `Total_Area` or similar.

Note names don't have to be long. A short abbreviation, as you might use in mathematics, is acceptable, if accompanied by a comment to indicate what is stored in the variable (e.g. A for area would be acceptable whereas using x for area would not).

It is also ok to follow standard conventions for loop variables (e.g. using i and j for loop variables)

Unnecessary code repetition avoided (5)

Loops and/or helper functions have been used to avoid unnecessary repetition of lines of highly similar code within a function.

Here is an example of unnecessary repetition. Suppose you were asked to display the first 5 square numbers and wrote the following script:

```
disp(1^2)
```

```
disp(2^2)
```

```
disp(3^2)
```

```
disp(4^2)
```

```
disp(5^2)
```

It works but has many highly similar lines repeated. This repetition could have been avoided by using a loop:

```
for i=1:5
    disp(i^2) end
```

Functionality (25 marks)

Functionality marks are awarded for each of the ten specified functions. There are 3 marks available per function (with the exception of ReadCipherImage, which is worth 4).

To get the functionality marks associated with a function your code must work EXACTLY as specified in the project document including the number, type(s) and order of any specified inputs/outputs.

Your functionality marks will be immediately scored when you submit your code for that function to Matlab Grader. E.g. if you submit your code for ReadCipherImage and earn 50% for that function, you will receive 2 out of the possible 4 points for that function.

You can submit each function up to **6 times**.

Before you submit your code we STRONGLY recommend you thoroughly test it and also use the “run function” area on Matlab Grader to check you have copied and pasted your code correctly (you don’t want to waste a submission for a simple copy/paste error) .