

Name : _____

Class Number: _____

ENGR 11, Laboratory 4: Initialize your Documentation Systems

Purpose of this lab:

More to be written here.

1. Create shortcuts for m-files with dynamic, documented headers

In this problem, you learn how to download code written by other MATLAB users that is available on [MATLAB's File Exchange](#). Specifically, you download the [New m-file \(dynamic/-documented header\)](#) written by [Dr. Benjamin Pillot](#). This code includes a MATLAB function file titled `new.m` that “creates a new script/function/class file following the same behavior as the ‘new file’ desktop short-cut, except that headers are dynamically implemented, customizable, and adapted for further code documentation.”

Recall that in our [Engr 11 Laboratory 2](#), you spent time reading the article [Best Practices for Scientific Computing](#) by [Greg Wilson et. al.](#) That paper featured a Box 1: Summary of Best Practices including the following items:

1. Write programs for people, not computers
 - (c) Make code style and formatting consistent.
4. Don't repeat yourself (or others)
 - (c) Re-use code instead of rewriting it.
7. Document design and purpose, not mechanics
 - (a) Document interfaces and reasons, not implementations.
 - (b) Rather than write a paragraph to explain a complex piece of code, reorganize the code so it doesn't need such an explanation.
 - (c) Embed the documentation for a piece of software in that software.

One of my goals as an instructor is to help you learn how to write code with these best practices in mind. A great place to is to systematize your processes for writing MATLAB documentation. That is exactly what you do in this problem.

- A. Navigate to the [MATLAB Central](#) homepage. Then, click on the [File Exchange](#) link at the top of the page (right next to the MATLAB Answers link).
 - (1.A.i) Find and click on the [About File Exchange](#) link.
 - (1.A.ii) Read the [About File Exchange](#) article.
 - (1.A.iii) Navigate back to the [File Exchange](#) landing page.
 - (1.A.iv) Find and click on the [Files](#) link.
 - (1.A.v) Explore the menu on the left-hand side of this Files page.

(1.A.vi) Find the “**Filter by Category**” section.

- Click the “Language Fundamentals” category. Spend 3 minutes exploring.
- Click the “Data Import and Export” category. Spend 3 minutes exploring.
- Click the “Mathematics” category. Spend 3 minutes exploring.
- Click the “Graphics” category. Spend 3 minutes exploring.
- Click the “Programming” category. Spend 3 minutes exploring.

Note: Item 1.A.vi is meant to take you at least 15 minutes. Put a timer on. Spend some time looking at those libraries. There is a lot available to you. I recommend that you look at the most three most popular files in each category. Click on each file and read the descriptions. This is your chance to start thinking about what is available to you.

B. Find the File Exchange article entitled [New m-file \(dynamic/documented header\)](#)

- (1.B.i) Read the Overview tab for the [New m-file \(dynamic/documented header\)](#) link.
- (1.B.ii) Read the [New m-file \(dynamic/documented header\)](#) files.
- (1.B.iii) Place these files in the appropriate place in your [MATLAB Search Path](#).
- (1.B.iv) Add these commands to your Quick Access Toolbar.

2. Start Your Own Blogs

In this problem, you start your own blog as a method to share your MATLAB learning with the world. This is part of my effort to encourage you to write your programs for people, not computers. You got your first taste of the importance of good practices for documenting your code as you completed Lab 2 when you read the article [Best Practices for Scientific Computing](#) by Greg Wilson et. al. Professional programmers value the themes mentioned in that article. In fact, other authors have dedicated much effort to this same topic. A famous and classic book on this subject is titled [The Elements of Programming Style](#) by Brian W. Kernighan and P.J. Plauger.¹ Some of the ways that those authors encourage their readers to write programs for people include a focus on the following principles:

- Build on the work of others, instead of starting from scratch each time.
- Format a program to help the reader understand your work.
- Write clearly - don't be too clever and don't sacrifice clarity for efficiency.

One fun feature of [scientific computation](#) is that you solve applied problems. Specifically, the idea of scientific computation is to use computers to solve problems in diverse areas of Science, Technology, Engineering, and Mathematics (STEM). The applied nature of this undertaking entails that much of the code you write will be based on expertise that you've built in topics that lie outside of traditional computer science.

If you are privileged enough to study and solve a problem that matters to other people, you likely will build significant and valuable insights that are not-so-easy to come by. Assuming you share your work with others, well-documented code is necessary but probably not sufficient to help on-board other programmers to your solution techniques. Documentation is designed to make code easier to read but doesn't always give deep insights into how you solved your problem or the decisions you made when creating your code. Moreover, there are limits on the type of information that you should include as formal comments in any one of your programs.

For example, it's hard to include images, figures, diagrams, or other heuristic tools within documentation since most computer programming languages have stringent syntax rules that do not permit embedded images. Similarly, because a single line of code can represent the output of many months of intense thought, it is impractical to try to document via comments all the knowledge a reader might need to know to decipher this line.

One way to augment your capacity to document your code and create a support structure for other users is to develop a personal blog where you can share technical writing. You can use such a platform to share important milestone in your progress and also to create an online platform to help other people understand your work. If you think about it, this is exactly the purpose of [MATLAB's Documentation Help Center](#) and also the [MATLAB Central online community](#). In this problem, you begin your journey of developing and refining systems to document your code by starting your own blog.

¹ A remarkable feature of this book is that the author [Brian W. Kernighan](#) helped to develop [Unix](#) at [Bell Labs](#). This [Elements of Programming Style](#) book is modeled after a famous style guide entitled [The Elements of Style](#), written by Strunk & White, that explores principles of writing composition for the English language.

-
- A. Read the article entitled [How to Make a Free Blog with WordPress: Set up your presence on the web in minutes](#) by Susan Gunelius (last updated on March 13, 2020).

I do not require you to use WordPress here. You can use any blogging platform you like. If you already have a blog that works, you're done with this part of problem 2. After having spent tens of hours researching blogging platforms, I whole-heartedly recommend using WordPress for a number of reasons. First, they provide a way to host your account for no out-of-pocket cost. Second, the platform is relatively intuitive and easy to use. Third, because there are a lot of people who use Wordpress, there is substantial online support and an avid community of users. These structures ensure that it's easy to get your questions answered with a [DuckDuckGo search](#) and a few clicks. Fourth, WordPress offers many tools to empower you to write your own [HTML code](#) and to include custom [CSS](#) code. Finally, I have some experience using WordPress and I might be able to help you if you get stuck. I cannot say the same for other blogging platforms.

- B. If you decide to use WordPress, then visit the [WordPress's platform](#). Follow the steps from the article in Lab 4, Problem 2A above and create your own blog.

- (2.B.i) Draft an "About" page with a brief introduction to yourself as a learner. To earn credit on this item, I am looking for a single paragraph. I encourage you to be honest and to think about this as a rough first draft. The important feature of this work is that you are learning how to do this, not that you have your work in a finalized form. If you are busy and have a lot to do in other spaces in your life, and spend no more than 15 minutes on your draft. Seriously. Put on a timer, write your draft, and move on. You can always come back and edit this later.

- (2.B.ii) Draft a one-paragraph introduction to your blog. The same comments I wrote in 2.B.i apply here. To get full credit, I'm looking for a single paragraph. Of course, you're welcome to write more if you're looking to ★-level accolades on this assignment. But to satisfy my vision for B-level work, I ask is for a single paragraph for problem 2.B.i and a different paragraph of problem 2.B.ii.

- C. Find an example of a technical blog that makes you smile and that you want to emulate.

A great way to develop a sense of the type of blog you want to create is to find examples of other bloggers who make you smile. In this problem, find one example of a technical blog that you feel strongly about. Hopefully you feel inspired by this work and enjoy the aesthetics of the site. Some key questions to ask yourself as you search include: How do I feel about the layout of this blog? What do I like or dislike about the way this author structures their work? How long are the posts? What topics does this person write about? How might I use this as inspiration as I start to develop my own technical blog to support my learning? Once you find a resource that you believe provides a lot of value, please capture the URL, Blog title, and author. You will share these with me to get credit on this problem .

Start Your Own Blogs: A-Level Extension Problems

For students who want a challenge, below are two A-level extensions to problem 2 on this lab. In addition to building your own blog, you may find it very helpful to build some expertise on this subject. Both of the to-do items below are designed to have you explore more advanced topics for creating technical blogs.

- D. Find at least one good article that documents best practices for creating a technical blog.

Writing a technical blog is as much of an art as it is a skill. This winter quarter 2021 is the first time I am having my ENGR 11 students create their own technical blogs. I am actively looking for good articles on this subject for share with future generations of students. Please find at least one resource that you believe will be useful for my future students. The major questions to ask yourself as you search are: how do I feel about the information I find in this research? How does this resources effect the way I think about my own blog? How inspired and empowered do I feel after having read through this resource? Once you find a resource that you believe provides a lot of value, please capture the URL, article title, author, and publication date. You will share these with me to get credit on this problem .

NOTE: I have specific features I am looking for in such articles. I am not interested in commercial sites that are trying to sell products or services. I don't want direct marketing for domain hosting or blog service. I am looking for good guides on best practices for hosting a technical blog. These should be relatively easy-to-understand, provide insights into how to structure a technical blog, and provide ideas about how to develop an online presence.

- E. Figure out how to embed [LaTeX code in your WordPress blog](#).

As we dive deeper into this course, I think you'll find that learning to code involves strong mathematical reasoning and lots of mathematical notation. I like to say that all computer scientists are mathematicians but not all mathematicians are computer scientists. Because so much learning and coding in MATLAB is based on developing deep intuition about mathematical ideas, it can be very helpful to know how to type mathematical notation. \LaTeX is a software designed specifically to communicate and publish scientific ideas that rely on mathematical notation. You can learn much more about \LaTeX by reading [about the LaTeX project](#). I believe it is possible to embed \LaTeX commands in your WordPress blog without having to have a working [TeX distribution](#) on your local machine.

To show that you have completed this item, I encourage you to write your first blog post on the topic of Proof by Induction. Specifically, you can submit any (or all) of your solutions to Lab 4, Problem 3 as a URL to a blog post on the topic of Mathematical Induction. In that post, you can use \LaTeX source code to document your work and communicate your ideas.

3. Learn How to Prove by Induction

In this problem, you will prove the following mathematical statements by induction:

$$\square \sum_{k=1}^n k = \frac{n(n+1)}{2}$$

$$\square \sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\square \sum_{k=0}^{n-1} 2^k = 2^n - 1$$

$$\square \sum_{k=1}^n k^3 = \frac{n^2(n+1)^2}{4}$$

More here.