Programming Project Submission Requirements

All figures and output (including tables) in the report for a given problem should be produced by a single program with a single execution. For example, if the project has three problems, there should be three executables (they can use other files as dependencies of course).

In the readme file, you should include a list of the figures and tables in the report, and the corresponding relative path and file name for each output (it is ok for the command line to print a table as long as you mention this also and output the table title in the command line as well).

Programming environments:

While all high-level language programming environment is allowed, we recommend you use the following.

Python on a local machine, Python on the ODU HPC cluster, Python on notebooks, (local notebooks or Jupyter notebooks on the HPC cluster) and Matlab submissions.

All environments:

- As a rule, I should not need to edit any code to run it. Consider the following:

  No absolute paths (e.g. C:/Users/khan/Documents/ digital_image_processing_project/problem_1.py would not be acceptable code in any program file)

  It is not allowed to have code that needs to be edited to change values to produce different figures, every value that needs to be tested should be hardcoded (and if there are large number of cases, a loop should be used).

- A readme file with all information needed to run the code needs to be included.
- All of these environments are easy to run, but the following needs to be done.

Python on a local machine:

Be sure it runs on Google Colab (this is straightforward and ensures your library dependencies are portable) if you are unsure of how to use Google Colab contact **in advance** of the deadline to be sure your code is acceptable and can run on this environment. This is required because Google Colab has a current and comprehensive library base, and therefore we can avoid deprecated libraries or unusual library combinations that do not mix well.

As an exception, if we need to develop a GUI, and you choose to use a local python code, use anaconda and provide an anaconda environment. To do this, we should be able to run the following and expect it to work for example:

conda create -n digital_image_processing_project python numpy pandas

Please test that you can create the environment in one create statement with all the packages listed. Of course, when you develop, installing packages in the middle of the coding will be needed, for example:

conda activate digital_image_processing_project python

writes code…

conda install numpy

writes more code…

conda install pandas

writes more code…

However, you **need to test that the following works**, because we can expect more compatibility with our system if you check that the environment can be built in a **single statement,** and you will need to provide this statement in your readme file for submission. (for more complex package dependencies installing one at a time versus building up front can give different behavior in anaconda)

conda create -n digital_image_processing_project python numpy pandas

Also, we don't want to create environments in multiple statements up front, as again a single statement gives the most predictable behavior. Avoid anything like the following when submitting (of course this may be useful when developing):

conda create -n digital_image_processing_project python

conda install -n digital_image_processing_project numpy

conda install -n digital_image_processing_project pandas

Of course, to test this is working, you will need to create an anaconda environment that is separate from the environment you developed the project in. You will need to copy your code into this new environment, collect a list of packages, guarantee that anaconda can build this list from a single create statement (as described above), and also guarantee that your code runs correctly with this build.

Python on the ODU HPC cluster:

The readme file should indicate which cluster this runs on (i.e. Turing or Wahab).

A bash script (or any shell) should be provided to load the required libraries.

So the following commands should work to run your code for problem 1 after I paste it into the folder "workspace":

cd "workspace"

salloc

source ./load_dependency_script.sh

python problem_1.py

Of course, if you need a different environment (e.g. gpu), you will need to provide the appropriate salloc command in your readme file.

Python on notebooks:

If you are running a notebook locally, test it on Google Colab (see the instructions for Python on a local machine above).

If you are running on the HPC Jupyter, I should have no problem running your notebook.

Matlab:

Include the version of Matlab in the readme along with any Matlab libraries used from the Matlab addons.

Other environments (C++, Java, etc.):

If you plan on using any other development tools, you need to discuss **in advance** your plan for making the code portable to my system so I can give feedback.

GitHub code, Matlab file exchange code, and any library or existing code should be approved by Dr. Iftekharuddin. This will be explained in the project requirements for each project and in class.

Project submission email: **mrahm006@odu.edu**