# EXPERIMENT 4: Soil Classification

PURPOSE

When soil borings are conducted as part of a subsurface investigation, a site engineer or geologist records the types of the soils encountered and the depth at which they were encountered. This information is recorded on a record or log of the boring. The field engineer must be able to accurately classify and describe soil samples in the field using visual and manual techniques, without the aid of laboratory equipment or laboratory tests. However, a reliable classification requires laboratory measurements of the grain-size distribution and Atterberg limits. Therefore, field classification are confirmed (or modified) based on subsequent laboratory tests. The Unified Soil Classification System (USCS) is the most commonly used classification system in Geotechnical Engineering practice across the US, and even internationally. The USCS classifies the soil based on the results obtained from particle size distribution (obtained from sieve and hydrometer tests) and soil plasticity (obtained from Liquid and Plastic Limit tests).

Considering the large number of USCS classifications performed on a regular basis, it would be efficient to write a short computer program to perform the soil classification, rather than doing it manually every time. For this lab, you are given a framework for a proposed soil classification program using Python and you are required to design and compile the coding for the USCS flowchart.

TEST STANDARD

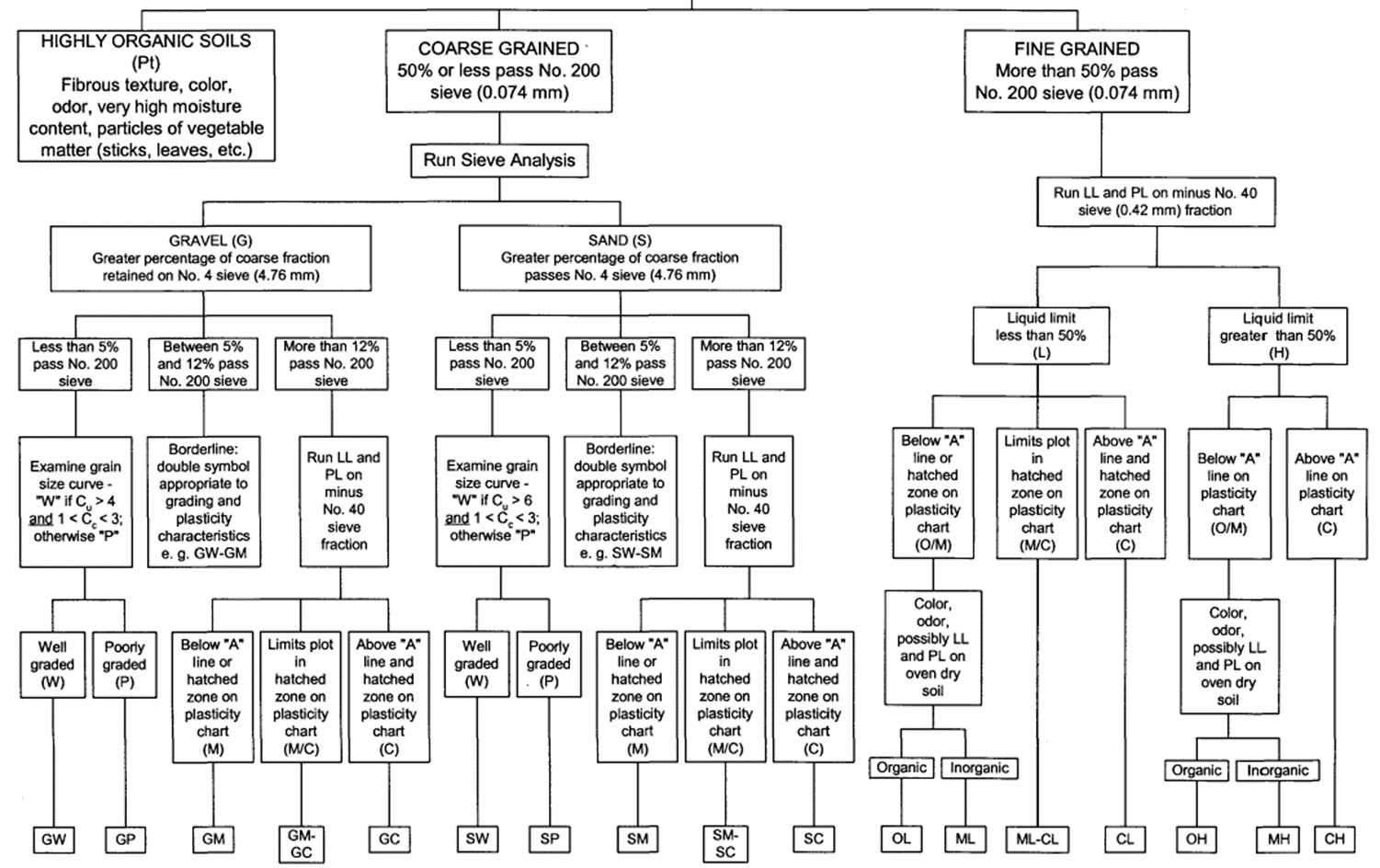
ASTM D2487–17 “Standard Practice for Classification of Soils for Engineering Purposes (Unified Soil Classification System)”

UNIFIED SOIL CLASSIFICATION SYSTEM

In the United States, classifications of soil in geotechnical engineering practice typically follow the Unified Soil Classification System (USCS). Two-letter symbols are used to indicate different USCS soil group types. The first letter describes the main constituent of the soil, while the second letter provides ancillary information regarding the grain-size distribution or plasticity of the soil. The group symbols and letters are given in Table 1, and should be familiar to you from discussions in class. Figure 1 details the USCS flowchart to identify the appropriate classification. Figure 2 shows the plasticity chart for classifying fine grained soils. For the purpose of this exercise, ignore the Organic soil classification.

Table 1. USCS Group Symbols

|  |  |  |
| --- | --- | --- |
| **Type of Soil** | **First Letter** | **Second Letter** |
| Coarse-Grained | G: Gravel  S: Sand | P: Poorly graded  W: Well graded  C: Clayey  M: Silty |
| Fine-Grained | M: Silt  C: Clay | H: High plasticity  L: Low plasticity |



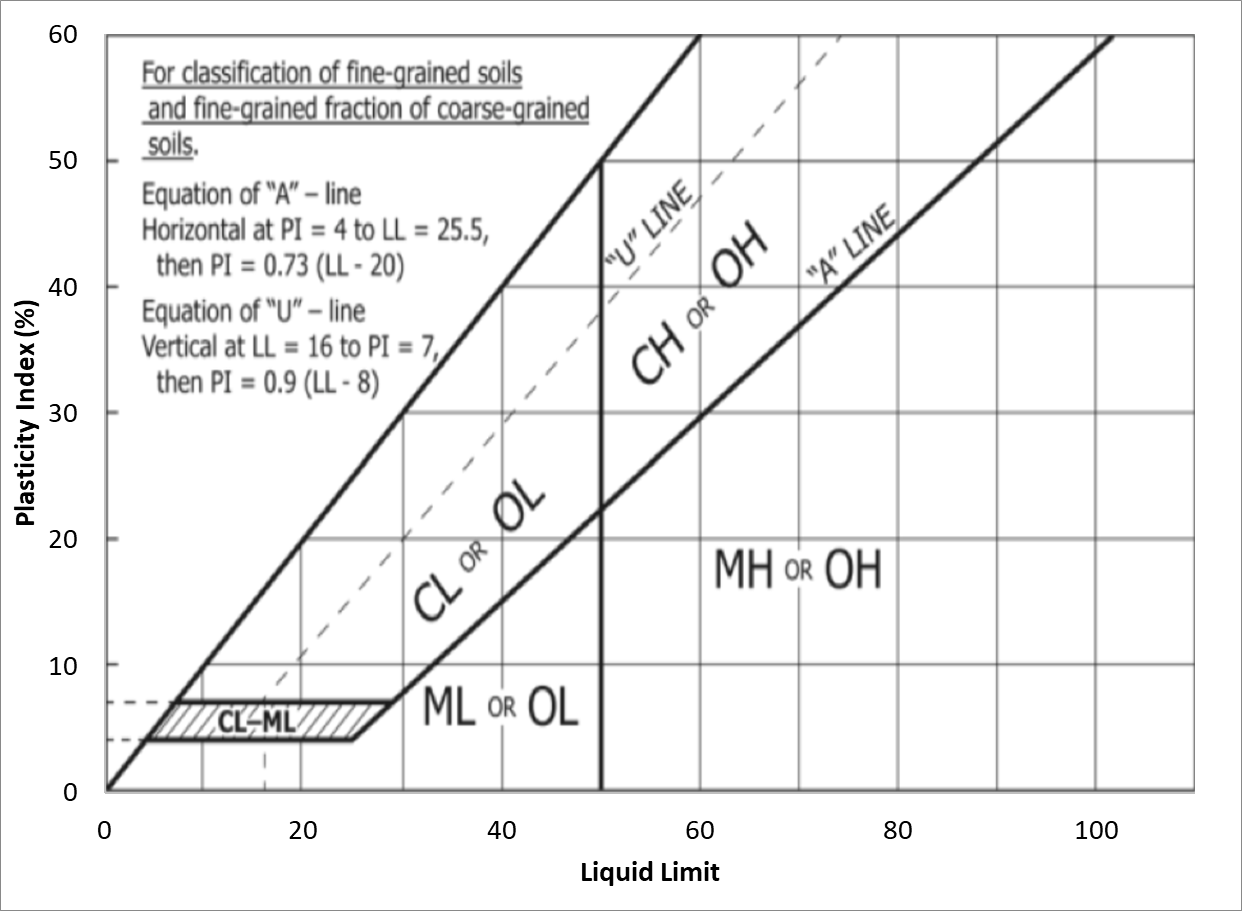


Figure 2. USCS Plasticity chart (after chart proposed by Casagrande)

PROGRAMING SOIL CLASSIFICATION WITH PYTHON

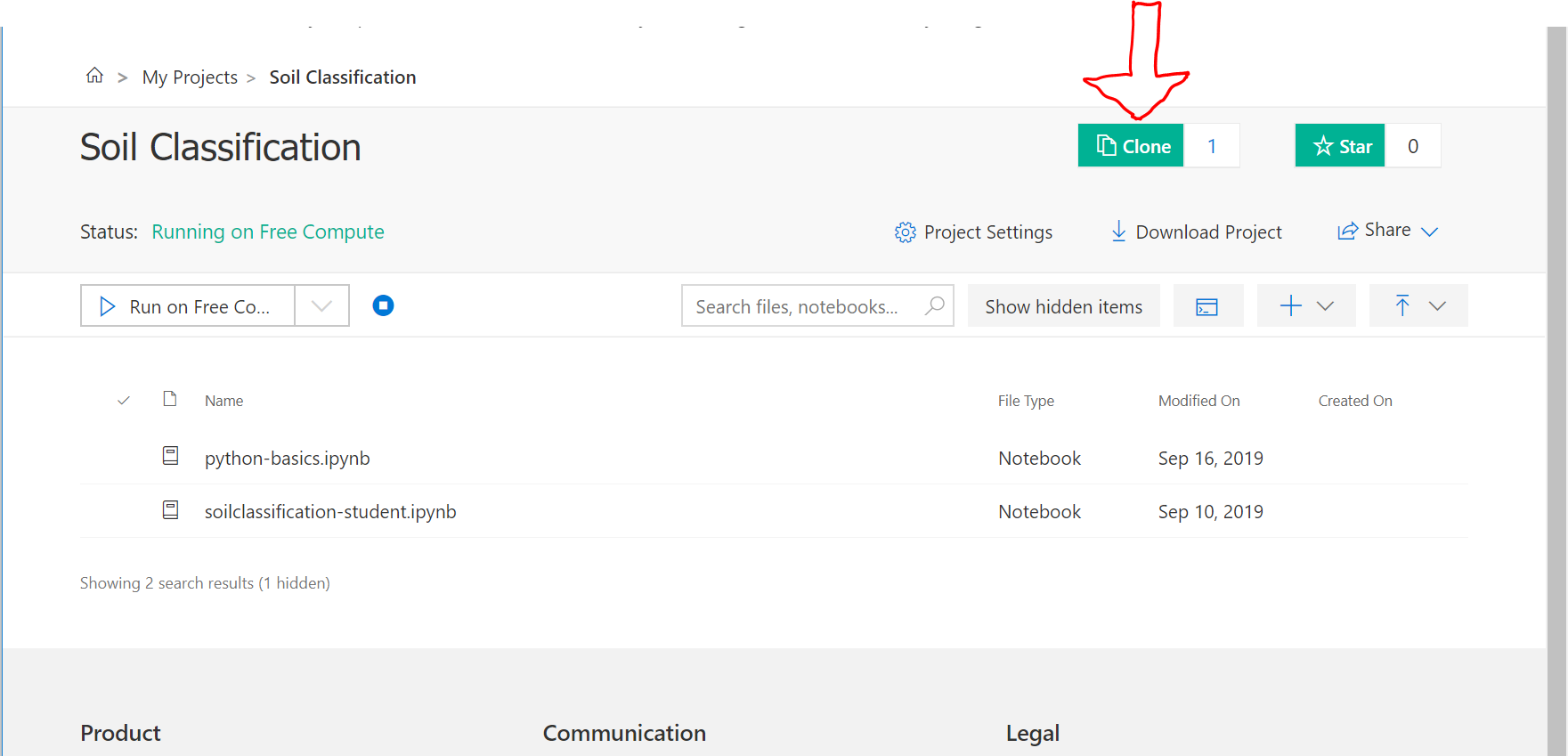
While many of you are not familiar with Python, the overall structure and thinking is the same for using any programming language. To ease you into the process, we will go through the basics of Python first and then get into the soil classification program

*Getting started*

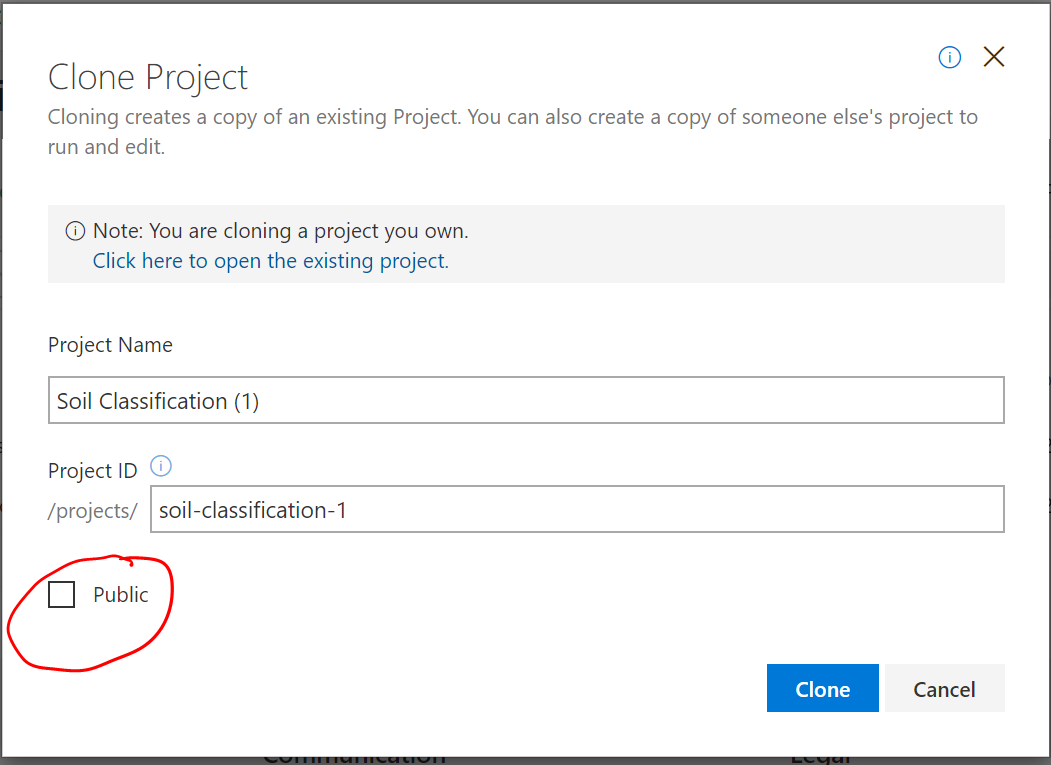
First step is getting you started on Running Jupyter Notebook on your laptops. We have created two files to help you with this assignment. The first file (python-basics.ipynb) is an overview of Python basics for those of you who never used it before and the second file (soilclassification-student.ipynb) is intended to help get you started on you classification programming.

To get started:

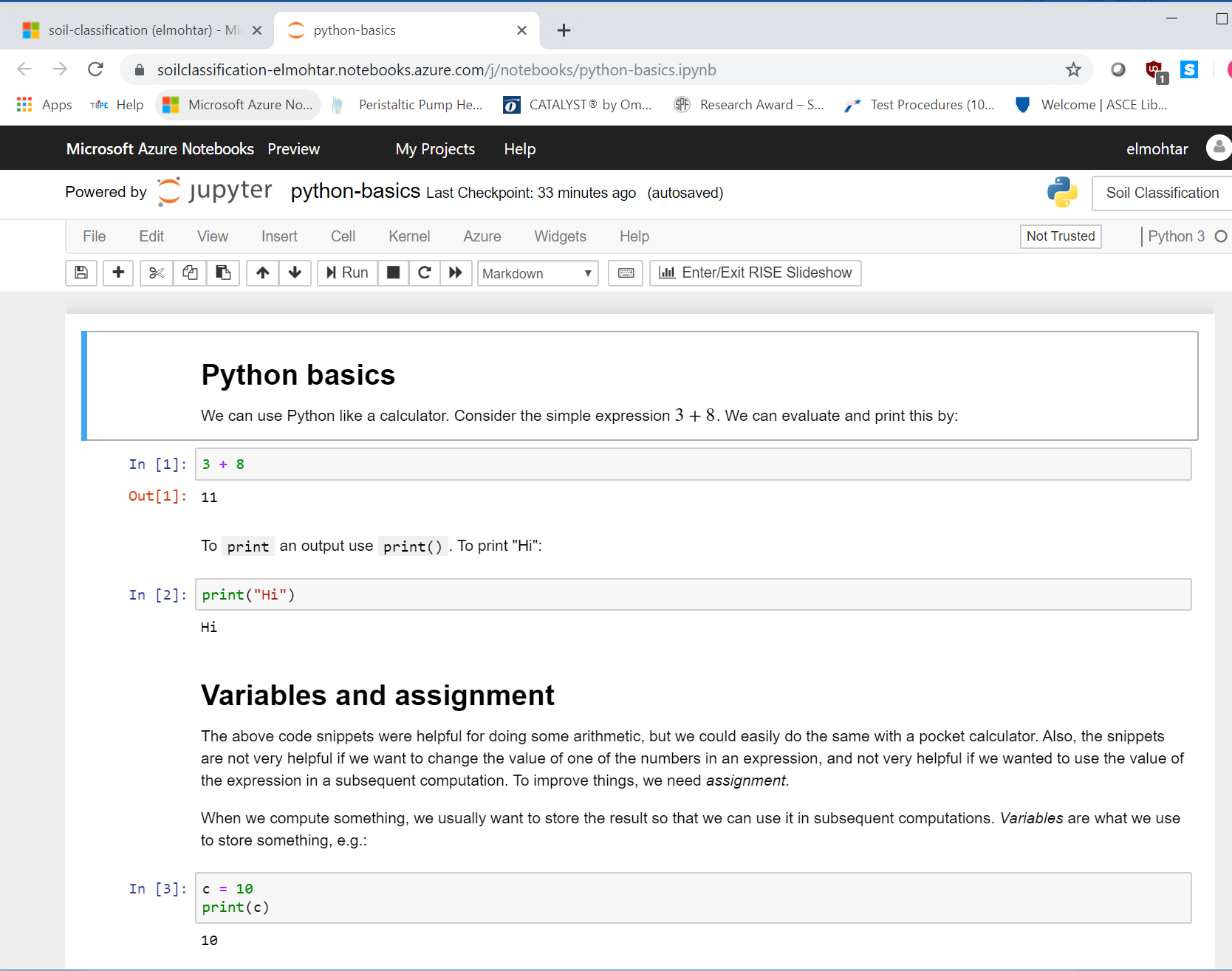
1. Sign in to <https://notebooks.azure.com/> using your UTexas email account.
2. Clone the project on soil classification by visiting <https://notebooks.azure.com/elmohtar/projects/soil-classification> and clicking on the clone button.



1. Clicking on the clone button will copy the soil-classification project to your My Project directory. Please ensure that you **DO NOT** click on the public option when cloning this project.



1. Once copied to your project, double click on the ‘python-basics.ipynb” file in the project folder. This will open the project. To run click on the “Run” button or Ctrl + Enter or CMD + Enter as shown below. Follow along the TA for a quick introduction that demonstrates the Python syntax on using variables and loops.



*Existing Framework*

Once you get familiar with using jupyter notebook, go back to the soil-classification project within your My Project directory and double click on the `soilclassification-student.ipynb` file. This will open the project. To run click on the “Run” button or Ctrl + Enter or CMD + Enter as shown below.

The initial part of the soil classification program (#Data input) has been provided for you. This part of the code will ask you to enter the required inputs to classify any given soil at the start of running the final program. **You do not have to modify this section of the code**. The input parameters used are defined as the following:

D10: 10th percentile of grain diameter (by weight), in mm

D30: 30th percentile of grain diameter (by weight), in mm

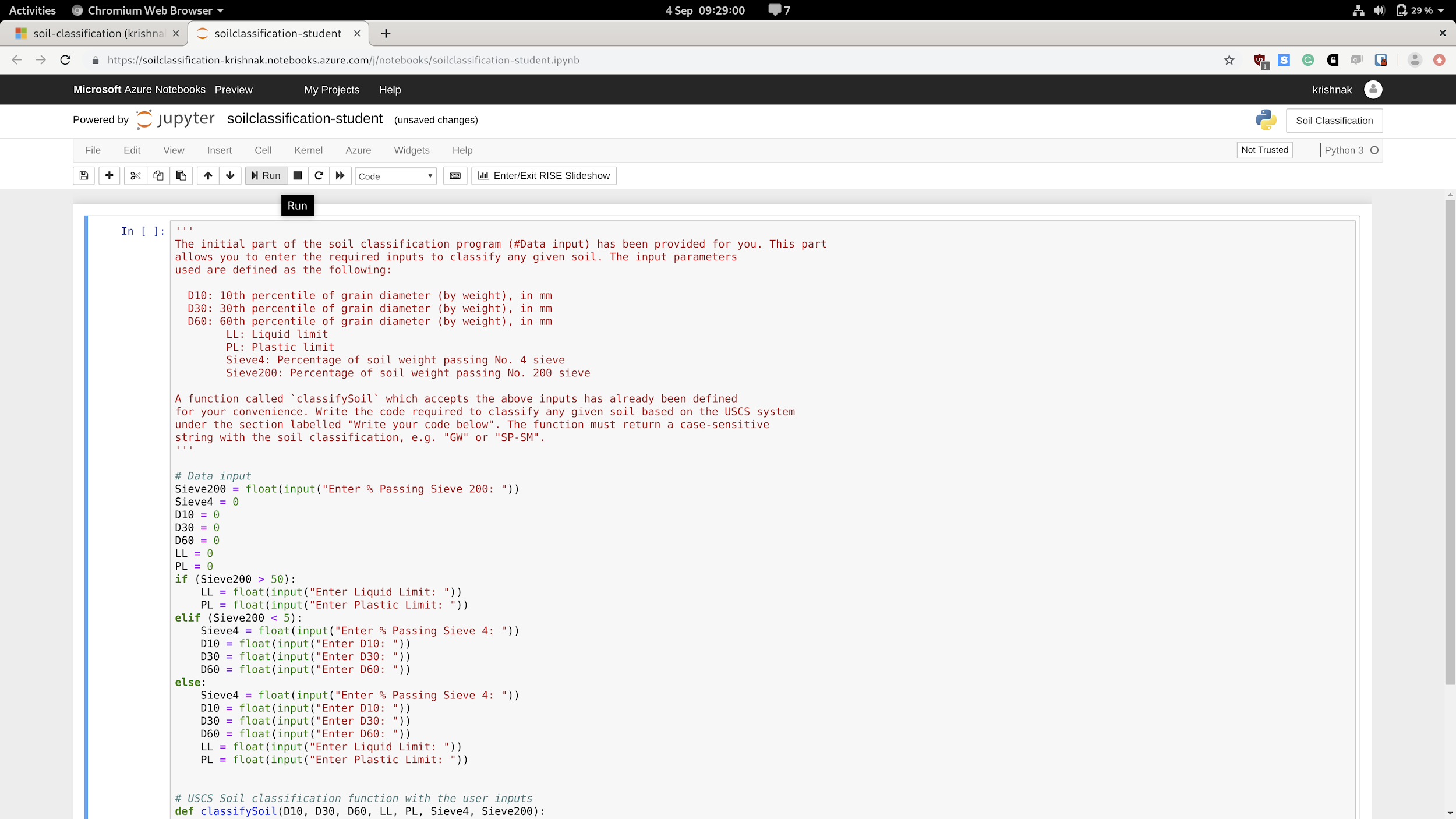
D60: 60th percentile of grain diameter (by weight), in mm

LL: Liquid limit

PL: Plastic limit

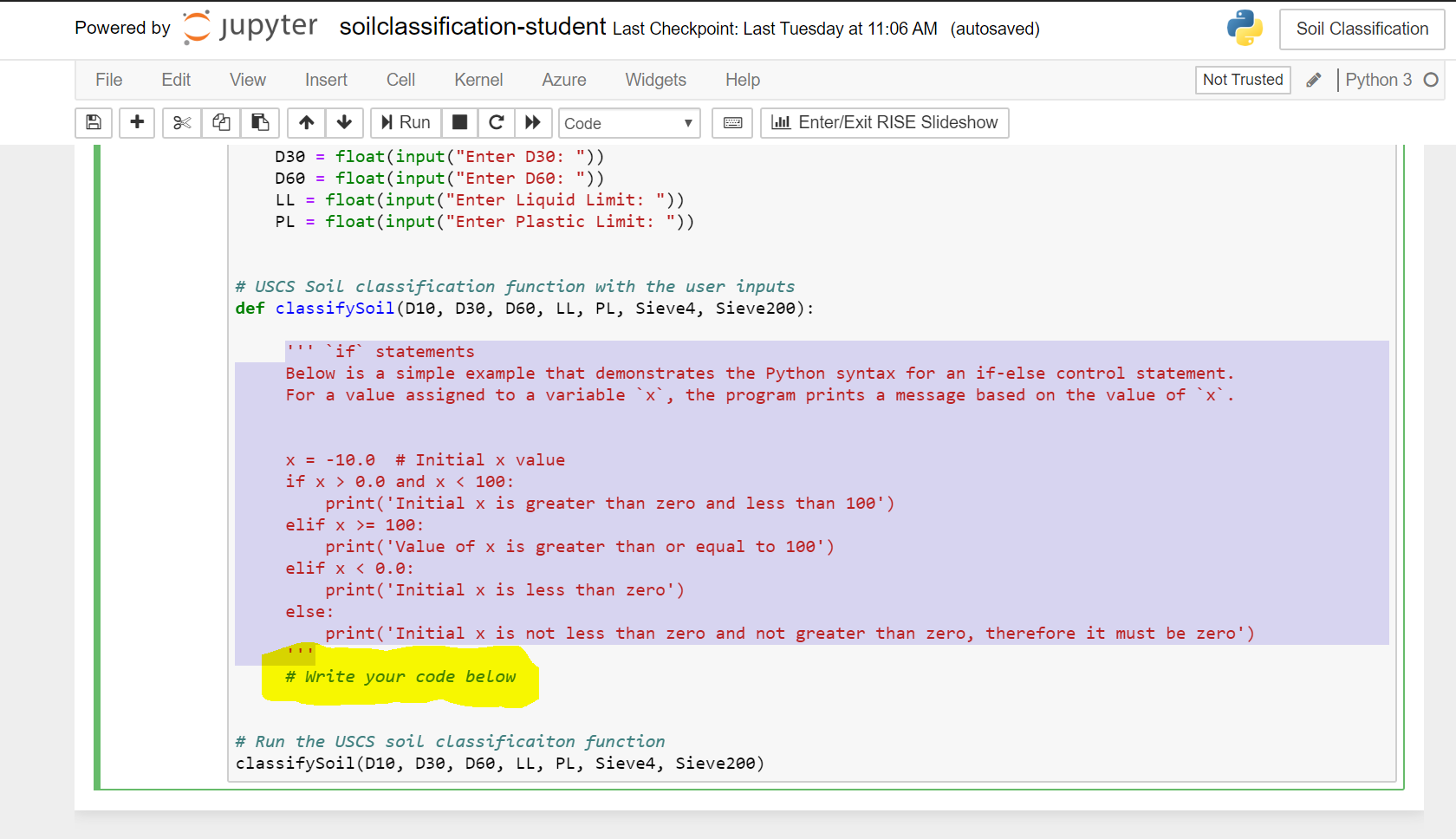
Sieve4: Percentage of soil weight passing No. 4 sieve

Sieve200: Percentage of soil weight passing No. 200 sieve



*Program Soil Classification*

A function called `classifySoil` which accepts the above inputs has already been defined for your convenience (see below). A simple example that demonstrates the Python syntax for an if-else control statement is also included for convenience. Write the code required to classify any given soil based on the USCS system under the section labelled "Write your code below". The function must return a case-sensitive string with the soil classification, e.g. "GW" or "SP-SM" based on the USCS classification flow chart in Figure 1 and Plasticity chart in Figure 2.



*Program Verification:*

Use the information for the three soils below to verify if your code is working properly. The appropriate classification for each soil is included in Table 2.

Table 2. Soil Classification for in-lab validation

|  |  |  |  |
| --- | --- | --- | --- |
| Soil | Soil I | Soil II | Soil III |
| D60 | - | 1.0mm | - |
| D30 | - | 0.3mm | - |
| D10 | - | 0.2mm | - |
| LL | 45 | - | 30 |
| PL | 15 | - | 23 |
| Sieve #4 | 85% | 70% | 60% |
| Sieve #200 | 55% | 4.5% | 25% |
| USCS Classification | CL | SP | GM |

*In-lab Submission:*

Before leaving the lab, save the ‘soilclassification-student.ipynb’ and download it to your computer and rename it ‘LastName-FirstName.ipynb’ and email it to your TA. Make sure to include in your email the USCS classification you obtained for Soil I, Soil II and Soil III. If you couldn’t finish the code to make it run, indicate that as well.

REPORT

1. Include a printout of the portion of the code that you were responsible for creating. In case you needed to make any changes after the conclusion of the lab (either because you didn’t get the right soil classification for Soil I, Soil II and/or Soil III or because you didn’t get to finish during the lab time), make sure to highlight these changes and explain them in the text.
2. Include the classification you obtained for Soil I, Soil II and Soil III.
3. Refer to Assignment 2 that you submitted for the class and classify Soils A through F using your program (input table is included below as well). Compare the classification you obtained to that obtained in the assignment and comment on the results.

Table 2. Table provided in Assignment 2.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Soil Sample | Percent Finer than | | | | | | | LL | PL |
| #4 | #10 | #20 | #40 | #100 | #200 | 0.002 mm |
| A | 75 | 47 | 27 | 16 | 8 | 3 | -- | Non – plastic | |
| B | 100 | 100 | 100 | 97 | 90 | 58 | 32 | 24 | 18 |
| C | 97 | 88 | 77 | 68 | 58 | 51 | 25 | 60 | 45 |
| D | 86 | 83 | 80 | 76 | 60 | 43 | 5 | 24 | 16 |
| E | 100 | 100 | 100 | 100 | 98 | 95 | 48 | 70 | 31 |
| F | 100 | 100 | 99 | 91 | 45 | 10 | -- | 42 | 34 |

1. After using the program to classify Soils I – III and Soils A – F, what recommendations do you have for changes to the code (either analysis or input parameters) to make it more useful for practicing engineers? Hint: consider the different inputs provided in Tables 2 and 3.

REFERENCES

ASTM D2487 (2017) “Standard Practice for Classification of Soils for Engineering Purposes (Unified Soil Classification System)”