Lab 6

Project 1.

Logging in to the HPC environment.

Open a FastX session to sp-udc-led-cs01.hpc.virginia.edu with the Eservices account assigned to you. Open a terminal.

This is a plain shell. Type
ls

Then type
mkdir aces
cd aces
mkdir lab6
cd lab6

Then type
module load gedit
You can type your programs into gedit. It is similar to Notepad++ on Windows or TextWranger on Mac.

Type a text file (any text file).

Start another terminal on the same desktop. Type
cd isscens/lab6

Then type
module load geany
geany &

Start another terminal. Type
module load anaconda
spyder &

When you feel comfortable logging in, you may shut down the terminals and FastX.

Project 2

A faculty member in the astronomy department has asked you to write a program for him to generate final exams for his introductory class. He provides you with the file testbank.txt which you can find in Resources/Data.

You are to write a program to randomly draw 25 questions from the test bank, then to scramble the multiple-choice answers. The answer choices will be written in the usual format of A, B, C, D, E. The program should also be able to grade the resulting exams.

Develop a document describing the functional requirements of the program.

Describe the design of your program. This should specify the data structures you have chosen to manipulate the data. Also describe any modules or classes you will write.

Describe the outputs and how they will be used when the program is in use.

Design a set of unit tests for the program. Note: you will need to "take" some of the generated tests in order to test the grading capabilities. You do not need to know anything about astronomy since the first answer (numbered 0) in the master file is always correct. Explain how your testing procedure will work.

Write and test the program. Use any language you want.

Check your program into SourceTree when you first start and be sure to commit all changes as you go.

Project 3

Exchange your test-generating program with another student using the same language. Conduct a code review of each other's code.

Project 4

Please read all instructions before beginning.

Standard chemical formulas order the elements in increasing electronegativity, e.g. HCl, $CaCO_3$. Another method called the Hill system: carbon first, followed by hydrogen if present, followed by the other elements in alphabetical order. If the

molecule has no carbon then all the elements, including H, are listed in alphabetical order. The Hill system is widely used for chemical databases.

Design and implement a program to convert a formula in one system into the other.

You will need at least three classes and possibly more (depending on your language and design). Create a class Element that contains the attributes Name, Symbol, and Atomic Number. The Element class should have a method print_element to print the chemical symbol given the atomic number. (Python: you may call this __str__ if you wish)

From the Element class derive a class Atom that extends Element with atomic mass and the value of the Pauling electronegativity. This class should overload addition (+) such that the "sum" of two atoms is the sum of their atomic masses. (Ignore any issues of mass deficit.)

Design a class Molecule that will use the Atom class. Suggestion: Molecule can be an array or list of Atoms. Molecule will contain the methods to convert standard (electronegativity) ordering to Hill ordering and vice versa. It will also have methods to compute and to print the weight of the molecule in Daltons.

We will not try to deal with subscripts so the digit indicating repeats will follow the elemental symbol. Your program does not have to handle formulas with parentheses. Feel free to try to permit them. If you do not allow parentheses be sure to reject the formula with an appropriate message.

The data required are in the file PeriodicTable.csv file in Resources/Data. Chemists will note that the atomic mass is only for the most abundant isotope. This is adequate for our purposes.

Test your code. Use at least

NaCl
CaCO3
NH4MnO4
C12H22O11