**ENGR 102 – Fall 2021**

**Lab Assignment #8**

**Deliverables:**

There are several deliverables for this team assignment. Please submit the following files to Canvas:

* Lab8\_Act1.pdf
* Lab8\_Act1\_XX.py where XX represents the initials of the team member
* Lab8\_Act1\_final.py
* Lab8\_Act1\_challenge.py (optional)

**Activity #1: Top-Down Design of a Program – to be done as a team**

Following the process described the lecture, as a team perform a top-down design for the program described below. **AFTER** you plan your program, write the code.

Open the Excel file entitled “Lab8-data.xlsx”. This document lists thermodynamic properties of liquid water at varying temperatures and at two different pressures. The properties listed are as follows:

* Specific volume () in units of
* Specific internal energy () in units of
* Specific enthalpy () in units of
* Specific entropy () in units of

It is common to use linear interpolation for temperature values not listed. So, for example, if you need the properties at , you could interpolate between the property values listed for and as a good estimate.

In your program, hard-code the temperature and property values for **as lists** for temperatures from to . Take as input a temperature in that range from the user. Your program should find the two values of temperature that bracket the user’s value, then perform linear interpolation for all four properties. The results should be formatted and printed to the screen based on the example output shown below. Print the specific volume, specific internal energy, specific enthalpy, and specific entropy to 7, 2, 2, and 4 decimal places, respectively.

**BEFORE WRITING ANY CODE**, create a document named Lab8\_Act1.pdf that will include parts a-c and g.

1. First, as a team, develop a top-down design for your program. Develop a hierarchy for the individual pieces, breaking each one down into as small of a piece so that the code for that portion of the program is “obvious” (about 5-10 lines of code). Put this hierarchy into your document. You may hand draw your hierarchy or use software to create it.
2. Next, as a team, determine what variables you will use for the main sections of your code. You should decide on the main data that you will need to keep that will be used in more than one “node” of the design. Write a list of these, along with a very brief (one sentence) description of what that variable will store. Note that you do not need to decide on variables that will be used only within one “node” of a program (e.g. you don’t need to describe a loop iterator if it is not going to be used outside the loop). Put this into your document.
3. As a team, discuss briefly the test cases you will need to test your code. Write down at least 10 test cases that you plan to use.
4. As a team, create one file with the comments/outline of the code. Share this file among all team members.
5. Next, divide the coding tasks among all team members, so that each person has a different section of the program.
   * Divide the “leaf” nodes among the team members so that each person has approximately the same number of items to implement.
   * If you have done a good job with the top-down design, and in specifying the variables that will carry over from one section to the next, then everyone should be able to write code for just their own section without seeing the other sections of the code!
6. Once each person has written their own separate code, bring the files together as a team, and have one person combine all of the code into one program.
   * You will likely need to thoroughly test and debug your code together as a team at this point. Use the test cases you developed in part c.
7. As a team, write a short summary (a few sentences – at least 75 words for each bullet point)
   * Describing the difficulty with which your team was able to combine the code at the end. Did this provide your team any insight into how the design itself might have been specified more clearly?
   * Describing any benefits and drawbacks you saw into dividing the coding like this. Can you see reasons why this might be a good idea? Can you see reasons why this might be a bad idea?

**As a challenge:** hard-code data for as well and develop an algorithm for double interpolation. Example: estimate properties at and . Name your file Lab8\_Act1\_challenge.py for up to **10 bonus points** on this assignment.

Things to submit:

1. Your document for parts a, b, c, and g as a single PDF file named Lab8\_Act1.pdf
2. The individual files you produced in part e named Lab8\_Act1\_XX.py where XX represents the initials for the team member that wrote the code
3. The single debugged file from part f named Lab8\_Act1\_final.py
4. The single debugged file for the challenge named Lab8\_Act1\_challenge.py (optional)

Example output (using input **50**):

Enter a temperature between 0 and 260 deg C: **50**

Properties at 50.0 deg C are:

Specific volume (m^3/kg): 0.0010103

Specific internal energy (kJ/kg): 208.60

Specific enthalpy (kJ/kg): 213.66

Specific entropy (kJ/kgK): 0.6996

Challenge example output (using input **50**, **7.5**):

Enter a temperature between 0 and 260 deg C: **50**

Enter a pressure between 5 and 10 MPa: **7.5**

Properties at 50.0 deg C and 7.5 MPa are:

Specific volume (m^3/kg): 0.0010092

Specific internal energy (kJ/kg): 208.24

Specific enthalpy (kJ/kg): 215.81

Specific entropy (kJ/kgK): 0.6984