Date 06/29/25

Lab 3

Mo Tu We Th Fr Sa Su

'Functions' chapter explained how to define and use functions in Python to make the code more organized, reusable, and easier to manage. A function in Python starts with the keyword 'def', followed by a name of our choice, parantheses (which can include input parameters), and a colon. Inside the function, we can write whatever task we want our program to perform. If we want the function to send back a result, we use the 'return' statement. For example, we can create a simple function to add two numbers and return the total, then reuse that logic throughout that program without rewriting the same code. I also learned that it's helpful to organize programs using a 'main()' function, and to include the signature line, "if __name__ == "__main__": main()" so that the code only runs when the file is executed. This helps avoid issues when importing code into other files. Another useful idea was that the functions can call other functions, which lets us break big problems into smaller parts — this is called decomposition.

A helpful way of building programs is called incremental development. Instead of writing a full function all at once, we can start with a basic version (called a stub), test it, and then gradually add more to it. This helps us catch mistakes early before things get too complicated.

As practice, we were asked to write a function that calculates the area of multiple pentagons, and to organize that into a seperate file that could be imported. This showed me how functions, decomposition, and clean code structure all come together in real programs.