Task 5 Part I:

I began the core implementation of the functions in Python as I was most familiar with that language. I found using 2 classes, one to store the line data and another to store all the records, was the easiest way to handle the csv file. Working in Python to figure out how to create each individual function made for an easier initial development of the solution as I could use many of Python's convenient searching tools like the keyword "in" to search through a string when deciding on which pieces of data had full values. Converting the Python code to Java was fairly easy as the two languages are both object-oriented languages which provided support for creating ClimateData objects to store the values. Java does have less searching capabilities though so I had to create a different method to determine if a row should be included in the set. The functions though were easy to replicate in Java as the logic for them was already figured out and both languages support easy concatenation of strings and string delimiting.

The C implementation was by far the most difficult of the 3 languages. Perhaps the most frustrating part of the C implementation is the lack of readable error messages or lack of error messages at all. Regarding the dynamic storage requirement the implementation is more complex compared to Python or Java as C requires that memory must be pre-allocated and reallocated whenever new data is to be added to create the dynamic list of data. Handling string concatenation and string delimiting is also more complex in C as the delimiter function in C "strtok" only returns the first entry of the row and as the input is a pointer, continuing the delimiting would cause the data string to change so a buffer string must be used in order to preserve the contents of the original data string. The actual logic for the C program is not so different from the logic of Python and Java as it does feature string comparisons like the other languages.

Task 5 Part II:

	Readability	Ease of Debugging	Development Speed	Execution Time for monthly report generation (sec) *	Lines of Code
Python	R1	R2	R2	~0.012	**310
Java	R2	R1	R1	~0.013	**392
C	R3	R3	R3	0.007	521

* Measurements for monthly report generation were based on generating the report for the average max gust speed for each month

** Lines of code were calculated using both the main program's lines as well as supporting class lines

~ Measurements were run multiple times to ensure validity and the general average time of the repeated trials was used as the final answer