­­­­­­Problem 1 - 15 points

When I think of computers I think of racing cars around a track. Using a computer how it is intended for normal consumer use is the same as buying a car off the lot and taking it to the track. Depending on how fast your car is you can put up a certain lap time. The more comfortable you get with your car the faster you can go, lowering your lap time. Learning how to take corners properly might be learning certain keyboard shortcuts to make things faster. But no matter what. Eventually your lap time will flatten out. This is because your car is stock and is made for efficiency and user comfort. If you want to get faster, you will have to start making modifications to your car. This is like writing programs to do tasks you would have to do manually. By putting faster parts and removing the parts you aren’t really using (turbo, better exhaust, removing the spare tire, radio, and getting better tires) you will lower your lap time. But you are also turning off traction control and autobraking which means you have to know what you are doing. Now, this is where a lot of people stop. If you want to go further you will have to build your own car (we will use Formula 1 cars in the analogy). Formula 1 cars are designed from the ground up with one goal in mind, get around the track as fast as possible. Since they have built this car from the ground up they can change anything they want to make it faster (ignoring FIA regulations). This means they can go much faster than your sports car off the lot and your modded sports car. But they require extensive training to be able to drive. This is like writing a program from scratch. You can do anything you want, but you have to do it all yourself.

Problem 2 - 15 points

I am Mechanical Engineering. I have already used python a bunch in testing procedures. I use an Arduino, which is in a version of C++, do collect data and monitor my testing procedures at work. This has saved me a bunch of time because while I am testing something for 5 days I can be doing something else. I also can take out human error from collecting data. It wasn’t easy to set it all up and that was a learning experience in itself. But once it was all set up it did exactly what I wanted it to do. I would monitor and record 15 different thermocouples, heater and fan wattage, put it all in a csv file, format that csv file so it was readable, and then it would graph the results so the data could be visualized. This alone saved me 20-30 hours of work, but it also took me 10 hours to write and debug. But now when I need to run a new test I can set it to run, put in my test information and parameters and let it run while I work on something else. I will admit I used chat gpt and my coworker knowledge to help me use libraries that I didn’t know how to use. But I will not be using chat gpt in this course.

Problem 3 - 15 points

I am always building stuff. Between my raspberry pi and my Arduino I am coding all the time. I also write programs to automate tasks on my computer. I am self-taught other than 1 programming class I took last semester at TCC. I know I have holes in my knowledge and I am excited to see where those are. I mainly use VS code but I am learning how to use VIM on my raspberry pi. I am also learning CLI.

Problem 4 - 5 Points  
Create an account on onlingdb.com. Then see if you can run the “Hello  
World!” Program. Add a line in the program showing your name and  
FSUID and take a screenshot. Paste the screenshot in your document.  
You don’t have to do this with your phone. Most computers come with  
a tool to take screenshots.

­

A screen shot of a computer

Description automatically generated