Lanny Oakman

Dr. Chong Yu

CS2028

04 September 2024

<div align="center">Lab 01 CS2028</div>

   Throughout the lab I was able to explore debugging techniques like breakpoints. Breakpoints allow for a deeper analysis of the variables at real-time by pausing the execution of the .cpp file at the indicated stop mark. Once the stop mark is reached, the user is able to then step through the code, understanding how and where the code runs. This is especially helpful for repeated actions like for loops with a nested if statement to help visualize the different cases. Another thing that was included in this lab was reworking previous code to utilize more preferred features (an array of structures vs several individual arrays). This process of fixing old code is important in my future career because it's very likely that I will be working on someone else's code. Being able to adapt and understand unfamiliar code that I didn't write is important in becoming a good programmer. In terms of this course I felt it was a good learning experience to play with and code structures. Classes/structures/object oriented programming in itself can seem difficult, so I am glad to get hands on experience in the classroom.
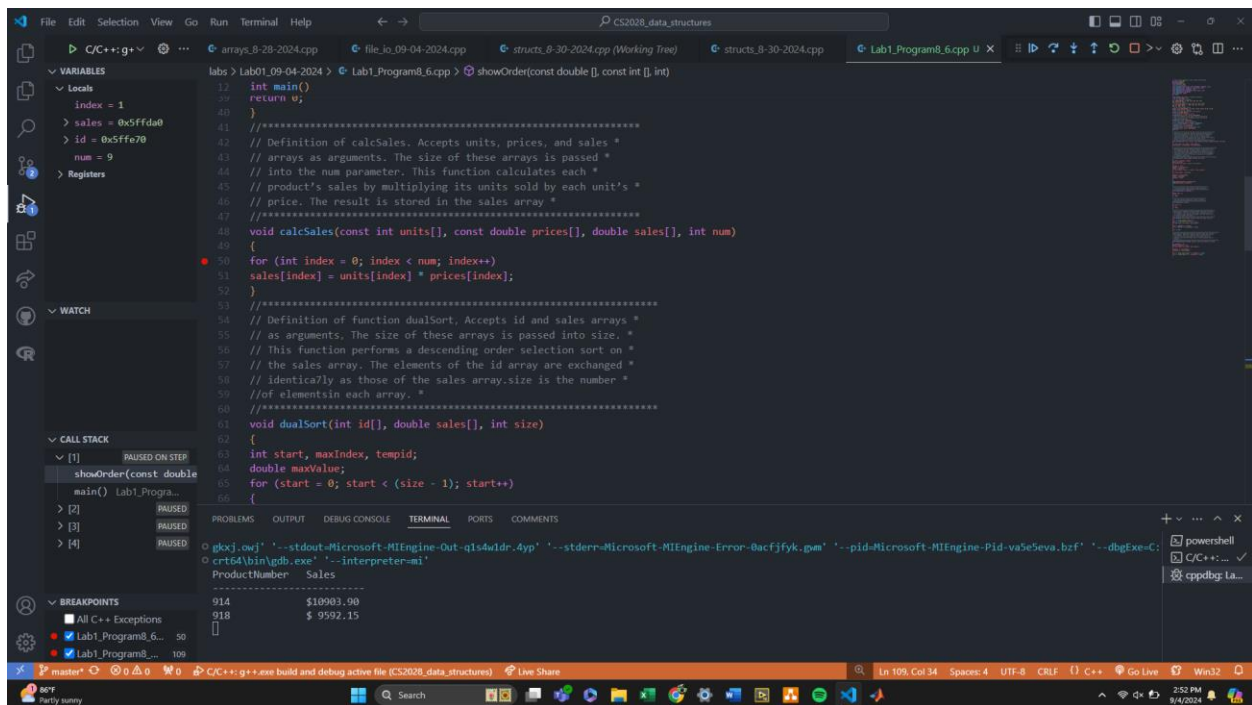

Fig. 1. Task 1 Screenshot

   As previously mentioned, the debugger was utilized to help resolve existing bugs in the .cpp file. I was able to step through the program and observe how the for loops all worked together. I believe that the programmer had made the mistakes due to a misunderstanding of how

arrays work and the lack of attention to accumulative variables placement. To avoid mistakes like this in the future, I'd recommend checking the console throughout the coding process to make sure that bad code isn't continually worsening due to a lack of care. I'd also recommend several test cases.
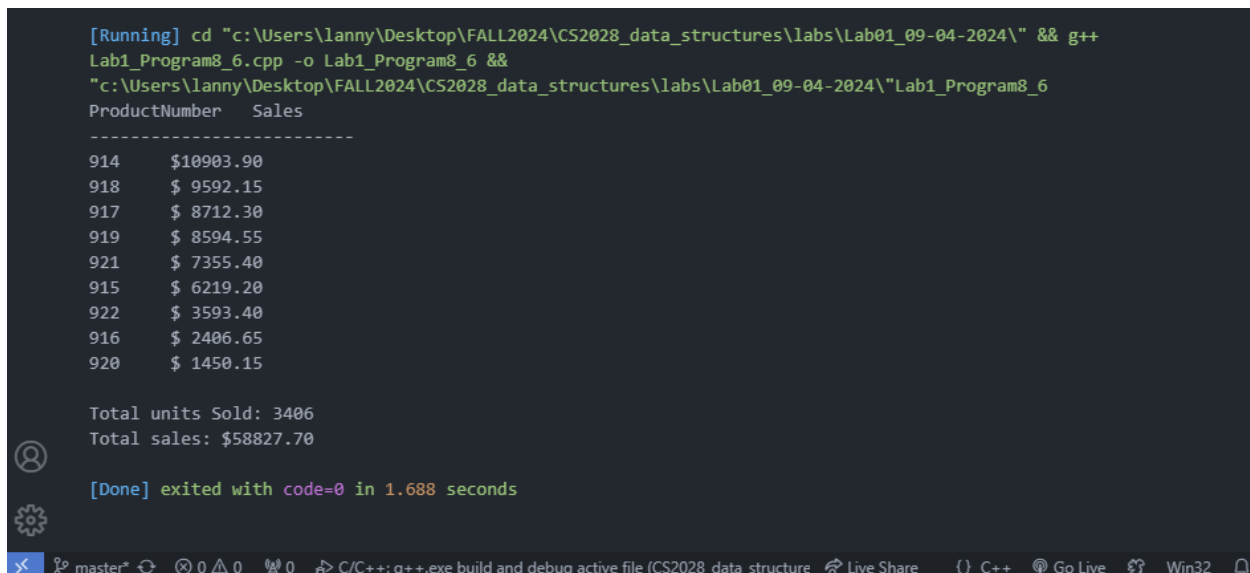

Fig. 2. Task 2 Screenshot

To complete Task 3, I first created a structure named product to hold an individual entity's information (id, units, prices, and sales). Once the structure was created I created a products array and added the previous data into the array of products. With the array of products created, I changed the function parameters to accept the product structure and the number of products constant. This introduced a series of bugs that resulted from not changing the existing function prototypes' parameters. I fixed this by changing the parameters to match the functions used in the main() function. After that, the program outputted as followed:

Fig. 3 Task 3 Screenshot