# Reflection Report on ChatGPT Usage

Cade Aihara

## Introduction:

This report about ChatGPT usage follows the lab assignment on ChatGPT. We were tasked with exploring how this AI software works and how it can be used in a constructive way. This report goes into depth on the tasks we were given as well as our commentary on the subject.

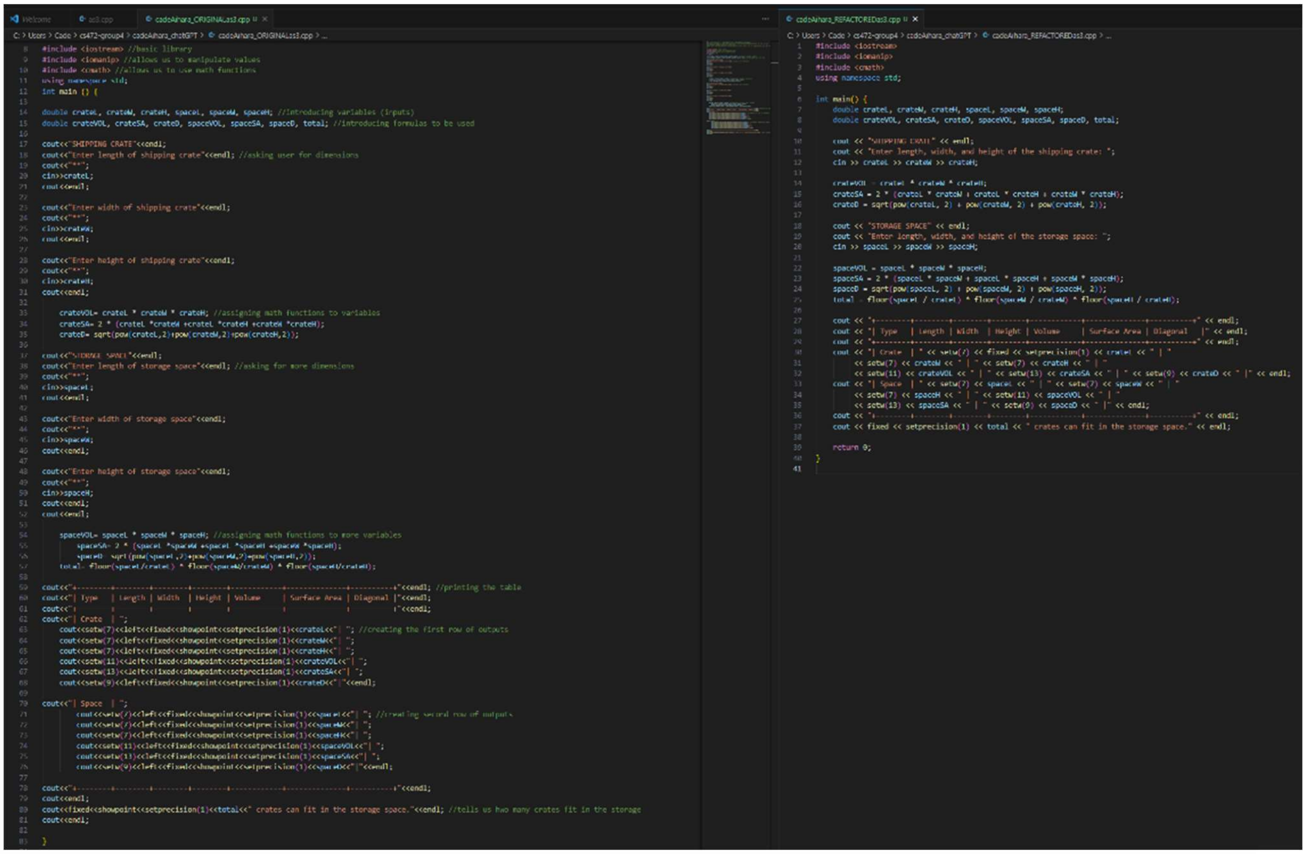
## Task 1: Familiarizing with ChatGPT:

This task primarily involved reading and playing with ChatGPT. As someone who has never used this software before, I was surprised to see how powerful this tool really is. I got carried away and started looking into how powerful it was because of how it grabbed my interest.

## Task 2.1: Code Refactoring:

For this task I chose to use one of my old CS135 assignments. I figured this was a good piece of code to use because of the simplicity of the function of the code and how inexperienced I was as a coder back then. I am aware that in my beginning stages of coding I was very inefficient in the way I coded and thought this would be a great opportunity to see how my code could be refactored.

When giving my code to ChatGPT and telling it to refactor it, I was impressed to see that it cut it down to half its length while maintaining the exact same functionality. I could see myself using this tool in the future as it may see areas of improvement quickly and making me a larger asset in the workforce. The code below is not meant to be read. It is meant to show the difference from original to refactored.

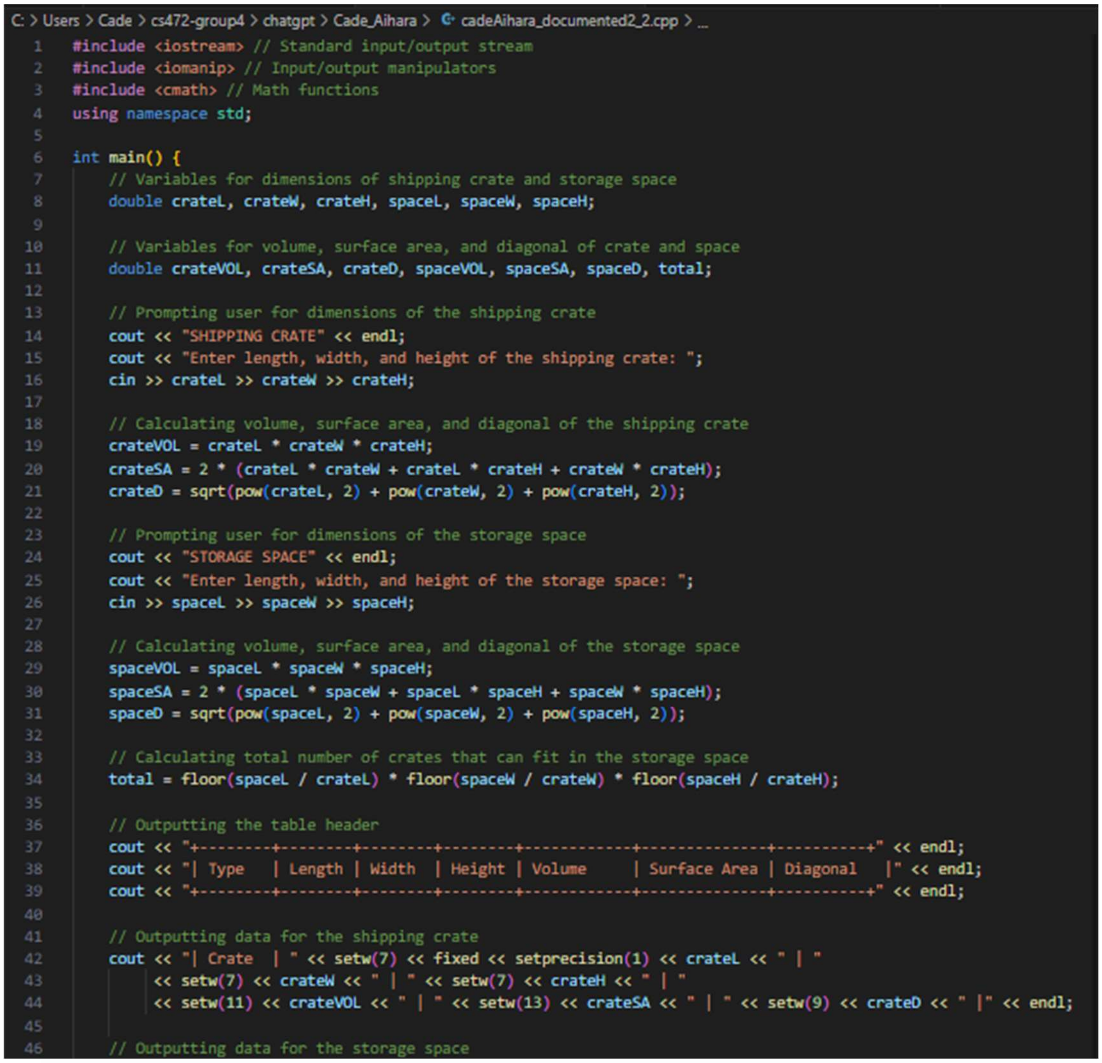

Welcome
as3.cpp
cadoAihara_ORIGINALas3.cpp
C: > Users > Cado > cs472-group4 > cadoAihara_chatGPT > cadoAihara_ORIGINALas3.cpp > ...
cadoAihara_REFACTOREDas3.cpp
C: > Users > Cado > cs472-group4 > cadoAihara_chatGPT > cadoAihara_REFACTOREDas3.cpp > ...

```cpp
#include <iostream> //basic library
#include <iomanip> //allows us to manipulate values
#include <cmath> //allows us to use math functions
using namespace std;
int main () {

double crateL, crateW, crateH, spaceL, spaceW, spaceH; //introducing variables (inputs)
double crateVOL, crateSA, crateD, spaceVOL, spaceSA, spaceD, total; //introducing formulas to be used

cout<<"SHIPPING CRATE"<<endl;
cout<<"Enter length of shipping crate"<<endl; //asking user for dimensions
cout<<"**";
cin>>crateL;
cout<<endl;

cout<<"Enter width of shipping crate"<<endl;
cout<<"**";
cin>>crateW;
cout<<endl;

cout<<"Enter height of shipping crate"<<endl;
cout<<"**";
cin>>crateH;
cout<<endl;

    crateVOL= crateL * crateW * crateH; //assigning math functions to variables
    crateSA= 2 * (crateL *crateW +crateL *crateH +crateW *crateH);
    crateD= sqrt(pow(crateL,2)+pow(crateW,2)+pow(crateH,2));

cout<<"STORAGE SPACE"<<endl;
cout<<"Enter length of storage space"<<endl; //asking for more dimensions
cout<<"**";
cin>>spaceL;
cout<<endl;

cout<<"Enter width of storage space"<<endl;
cout<<"**";
cin>>spaceW;
cout<<endl;

cout<<"Enter height of storage space"<<endl;
cout<<"**";
cin>>spaceH;
cout<<endl;
cout<<endl;

    spaceVOL= spaceL * spaceW * spaceH; //assigning math functions to more variables
        spaceSA= 2 * (spaceL *spaceW +spaceL *spaceH +spaceW *spaceH);
        spaceD= sqrt(pow(spaceL,2)+pow(spaceW,2)+pow(spaceH,2));
    total= floor(spaceL/crateL) * floor(spaceW/crateW) * floor(spaceH/crateH);

cout<<"+--------+--------+--------+--------+-----------+-------------+----------+"<<endl; //printing the table
cout<<"| Type   | Length | Width  | Height | Volume    | Surface Area | Diagonal |"<<endl;
cout<<"|        |        |        |        |           |             |          |"<<endl;
cout<<"| Crate  | ";
    cout<<setw(7)<<left<<fixed<<showpoint<<setprecision(1)<<crateL<<"| "; //creating the first row of outputs
    cout<<setw(7)<<left<<fixed<<showpoint<<setprecision(1)<<crateW<<"| ";
    cout<<setw(7)<<left<<fixed<<showpoint<<setprecision(1)<<crateH<<"| ";
    cout<<setw(11)<<left<<fixed<<showpoint<<setprecision(1)<<crateVOL<<"| ";
    cout<<setw(13)<<left<<fixed<<showpoint<<setprecision(1)<<crateSA<<"| ";
    cout<<setw(9)<<left<<fixed<<showpoint<<setprecision(1)<<crateD<<"|"<<endl;

cout<<"| Space  | ";
        cout<<setw(7)<<left<<fixed<<showpoint<<setprecision(1)<<spaceL<<"| "; //creating second row of outputs
        cout<<setw(7)<<left<<fixed<<showpoint<<setprecision(1)<<spaceW<<"| ";
        cout<<setw(7)<<left<<fixed<<showpoint<<setprecision(1)<<spaceH<<"| ";
        cout<<setw(11)<<left<<fixed<<showpoint<<setprecision(1)<<spaceVOL<<"| ";
        cout<<setw(13)<<left<<fixed<<showpoint<<setprecision(1)<<spaceSA<<"| ";
        cout<<setw(9)<<left<<fixed<<showpoint<<setprecision(1)<<spaceD<<"|"<<endl;

cout<<"+--------+--------+--------+--------+-----------+-------------+----------+"<<endl;
cout<<endl;
cout<<fixed<<showpoint<<setprecision(1)<<total<<" crates can fit in the storage space."<<endl; //tells us hwo many crates fit in the storage
cout<<endl;

}
```

```cpp
#include <iostream>
#include <iomanip>
#include <cmath>
using namespace std;

int main() {
    double crateL, crateW, crateH, spaceL, spaceW, spaceH;
    double crateVOL, crateSA, crateD, spaceVOL, spaceSA, spaceD, total;

    cout << "SHIPPING CRATE" << endl;
    cout << "Enter length, width, and height of the shipping crate: ";
    cin >> crateL >> crateW >> crateH;

    crateVOL = crateL * crateW * crateH;
    crateSA = 2 * (crateL * crateW + crateL * crateH + crateW * crateH);
    crateD = sqrt(pow(crateL, 2) + pow(crateW, 2) + pow(crateH, 2));

    cout << "STORAGE SPACE" << endl;
    cout << "Enter length, width, and height of the storage space: ";
    cin >> spaceL >> spaceW >> spaceH;

    spaceVOL = spaceL * spaceW * spaceH;
    spaceSA = 2 * (spaceL * spaceW + spaceL * spaceH + spaceW * spaceH);
    spaceD = sqrt(pow(spaceL, 2) + pow(spaceW, 2) + pow(spaceH, 2));
    total = floor(spaceL / crateL) * floor(spaceW / crateW) * floor(spaceH / crateH);

    cout << "+--------+--------+--------+--------+-----------+-------------+-----------+" << endl;
    cout << "| Type   | Length | Width  | Height | Volume    | Surface Area | Diagonal  |" << endl;
    cout << "+--------+--------+--------+--------+-----------+-------------+-----------+" << endl;
    cout << "| Crate  | " << setw(7) << fixed << setprecision(1) << crateL << " | "
         << setw(7) << crateW << " | " << setw(7) << crateH << " | "
         << setw(11) << crateVOL << " | " << setw(13) << crateSA << " | " << setw(9) << crateD << " |" << endl;
    cout << "| Space  | " << setw(7) << spaceL << " | " << setw(7) << spaceW << " | "
         << setw(7) << spaceH << " | " << setw(11) << spaceVOL << " | "
         << setw(13) << spaceSA << " | " << setw(9) << spaceD << " |" << endl;
    cout << "+--------+--------+--------+--------+-----------+-------------+-----------+" << endl;
    cout << fixed << setprecision(1) << total << " crates can fit in the storage space." << endl;

    return 0;
}
```

Task 2.2: Documentation Assistance:

For documentation assistance, I chose to build off of the refactored code from the previous task. For this task I prompted ChatGPT requesting documentation on my code to increase reader's ability to understand my code. It seemed to work perfectly and was able to move on rather quickly. Here is what the code looked like after documentation.



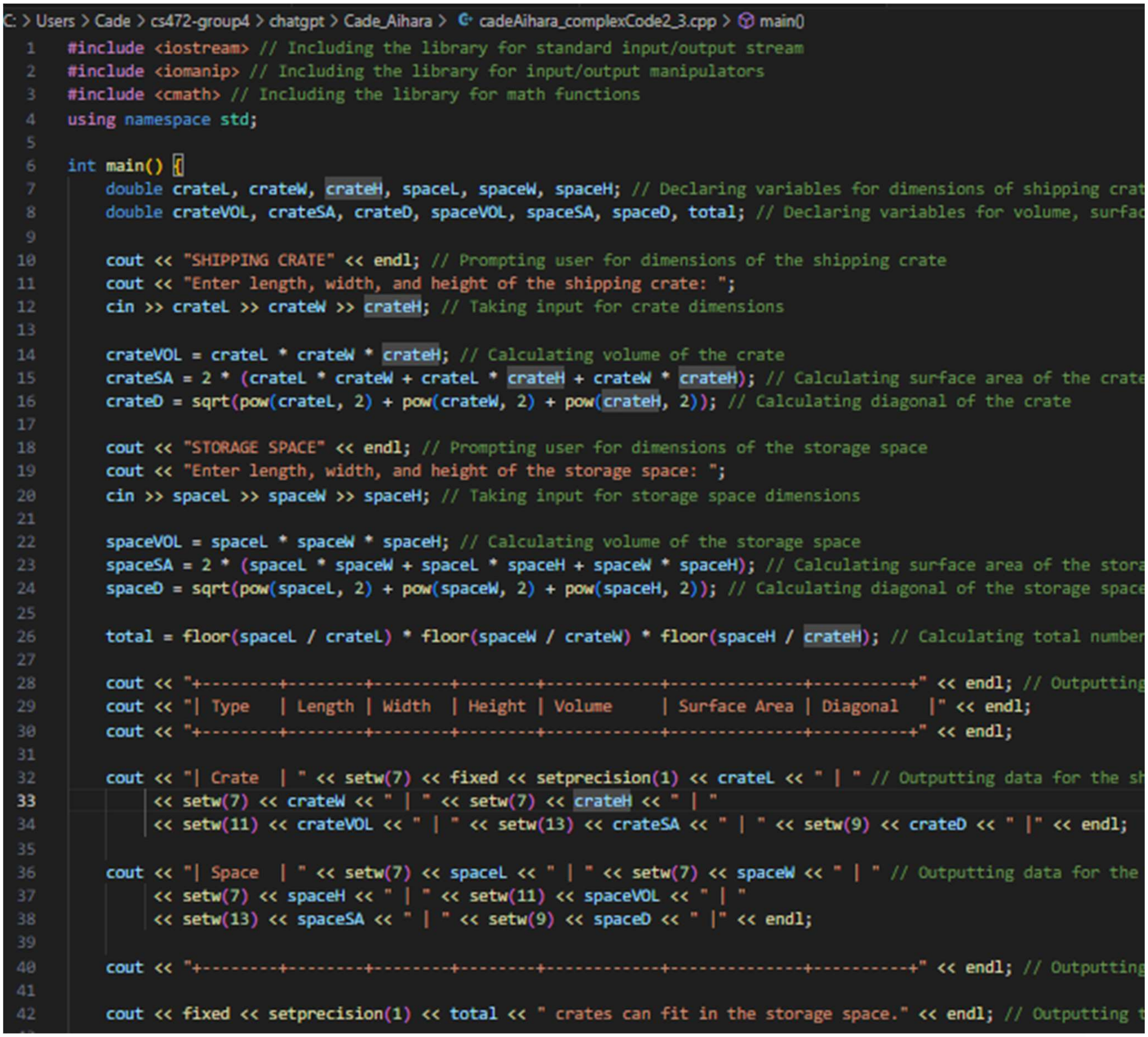
C: > Users > Cade > cs472-group4 > chatgpt > Cade_Aihara > cadeAihara_documented2_2.cpp > ...

```cpp
#include <iostream> // Standard input/output stream
#include <iomanip> // Input/output manipulators
#include <cmath> // Math functions
using namespace std;

int main() {
    // Variables for dimensions of shipping crate and storage space
    double crateL, crateW, crateH, spaceL, spaceW, spaceH;

    // Variables for volume, surface area, and diagonal of crate and space
    double crateVOL, crateSA, crateD, spaceVOL, spaceSA, spaceD, total;

    // Prompting user for dimensions of the shipping crate
    cout << "SHIPPING CRATE" << endl;
    cout << "Enter length, width, and height of the shipping crate: ";
    cin >> crateL >> crateW >> crateH;

    // Calculating volume, surface area, and diagonal of the shipping crate
    crateVOL = crateL * crateW * crateH;
    crateSA = 2 * (crateL * crateW + crateL * crateH + crateW * crateH);
    crateD = sqrt(pow(crateL, 2) + pow(crateW, 2) + pow(crateH, 2));

    // Prompting user for dimensions of the storage space
    cout << "STORAGE SPACE" << endl;
    cout << "Enter length, width, and height of the storage space: ";
    cin >> spaceL >> spaceW >> spaceH;

    // Calculating volume, surface area, and diagonal of the storage space
    spaceVOL = spaceL * spaceW * spaceH;
    spaceSA = 2 * (spaceL * spaceW + spaceL * spaceH + spaceW * spaceH);
    spaceD = sqrt(pow(spaceL, 2) + pow(spaceW, 2) + pow(spaceH, 2));

    // Calculating total number of crates that can fit in the storage space
    total = floor(spaceL / crateL) * floor(spaceW / crateW) * floor(spaceH / crateH);

    // Outputting the table header
    cout << "+--------+--------+--------+--------+------------+--------------+----------+" << endl;
    cout << "| Type   | Length | Width  | Height | Volume     | Surface Area | Diagonal   |" << endl;
    cout << "+--------+--------+--------+--------+------------+--------------+----------+" << endl;

    // Outputting data for the shipping crate
    cout << "| Crate  | " << setw(7) << fixed << setprecision(1) << crateL << " | "
         << setw(7) << crateW << " | " << setw(7) << crateH << " | "
         << setw(11) << crateVOL << " | " << setw(13) << crateSA << " | " << setw(9) << crateD << " |" << endl;

    // Outputting data for the storage space
```

Task 2.3: Understanding Complex Code:

This task of understanding complex code was a bit challenging. I had to rephrase my prompt when asking ChatGPT to do what I requested. It would often make comments on the code after it sent me the code back. This made the comments independent form the file which was undesired.

After playing around with the wording and phrasing of my request I was able to finally receive what I needed. It integrated the comments into the code and was happy with the result. My code was not very complex so there wasn't a ton to break down. However, I t was able to explain each of my code lines in detail further than the previous step.

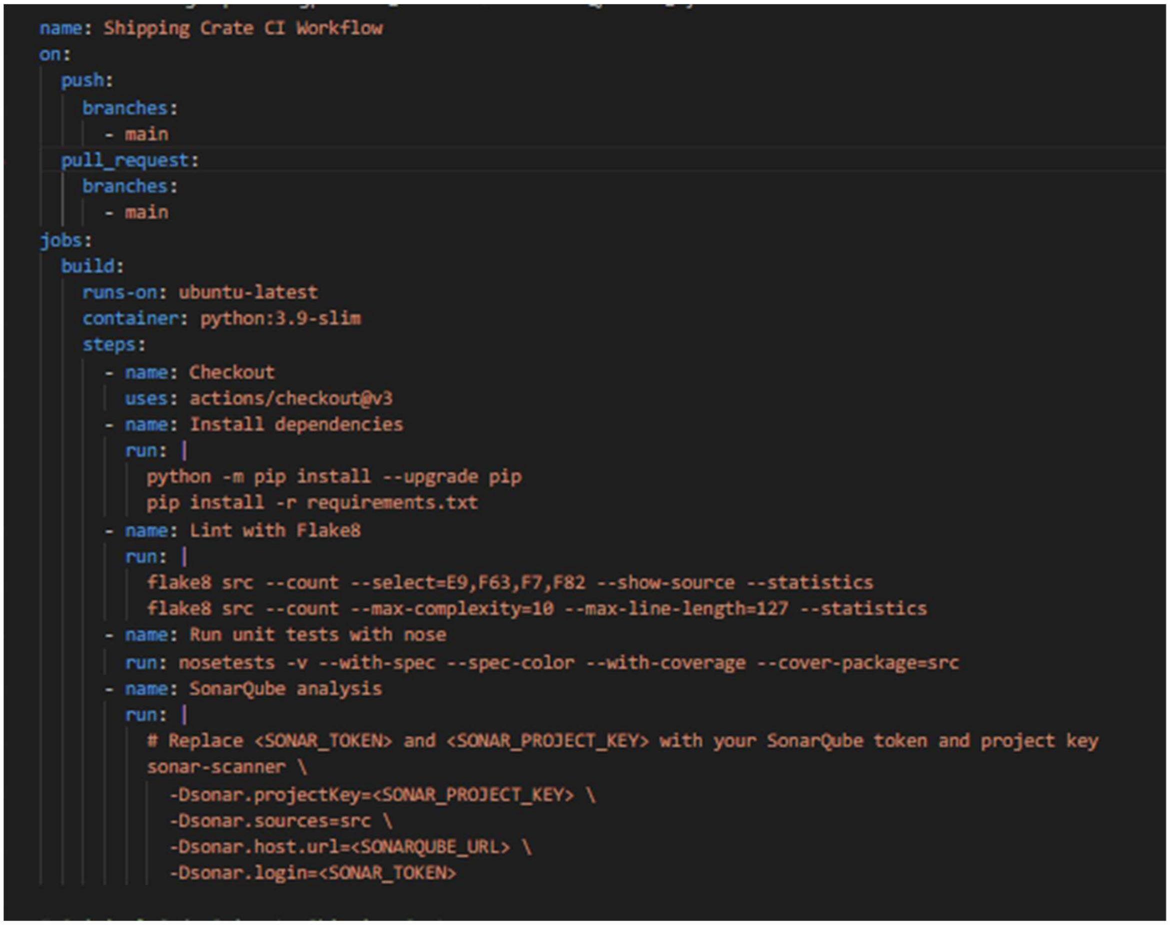


C: > Users > Cade > cs472-group4 > chatgpt > Cade_Aihara > cadeAihara_complexCode2_3.cpp > main()

```cpp
#include <iostream> // Including the library for standard input/output stream
#include <iomanip> // Including the library for input/output manipulators
#include <cmath> // Including the library for math functions
using namespace std;

int main() {
    double crateL, crateW, crateH, spaceL, spaceW, spaceH; // Declaring variables for dimensions of shipping crat
    double crateVOL, crateSA, crateD, spaceVOL, spaceSA, spaceD, total; // Declaring variables for volume, surfac

    cout << "SHIPPING CRATE" << endl; // Prompting user for dimensions of the shipping crate
    cout << "Enter length, width, and height of the shipping crate: ";
    cin >> crateL >> crateW >> crateH; // Taking input for crate dimensions

    crateVOL = crateL * crateW * crateH; // Calculating volume of the crate
    crateSA = 2 * (crateL * crateW + crateL * crateH + crateW * crateH); // Calculating surface area of the crate
    crateD = sqrt(pow(crateL, 2) + pow(crateW, 2) + pow(crateH, 2)); // Calculating diagonal of the crate

    cout << "STORAGE SPACE" << endl; // Prompting user for dimensions of the storage space
    cout << "Enter length, width, and height of the storage space: ";
    cin >> spaceL >> spaceW >> spaceH; // Taking input for storage space dimensions

    spaceVOL = spaceL * spaceW * spaceH; // Calculating volume of the storage space
    spaceSA = 2 * (spaceL * spaceW + spaceL * spaceH + spaceW * spaceH); // Calculating surface area of the stora
    spaceD = sqrt(pow(spaceL, 2) + pow(spaceW, 2) + pow(spaceH, 2)); // Calculating diagonal of the storage space

    total = floor(spaceL / crateL) * floor(spaceW / crateW) * floor(spaceH / crateH); // Calculating total number

    cout << "+--------+--------+--------+--------+------------+--------------+-----------+" << endl; // Outputting
    cout << "| Type   | Length | Width  | Height | Volume     | Surface Area | Diagonal  |" << endl;
    cout << "+--------+--------+--------+--------+------------+--------------+-----------+" << endl;

    cout << "| Crate  | " << setw(7) << fixed << setprecision(1) << crateL << " | " // Outputting data for the sh
         << setw(7) << crateW << " | " << setw(7) << crateH << " | "
         << setw(11) << crateVOL << " | " << setw(13) << crateSA << " | " << setw(9) << crateD << " |" << endl;

    cout << "| Space  | " << setw(7) << spaceL << " | " << setw(7) << spaceW << " | " // Outputting data for the
         << setw(7) << spaceH << " | " << setw(11) << spaceVOL << " | "
         << setw(13) << spaceSA << " | " << setw(9) << spaceD << " |" << endl;

    cout << "+--------+--------+--------+--------+------------+--------------+-----------+" << endl; // Outputting

    cout << fixed << setprecision(1) << total << " crates can fit in the storage space." << endl; // Outputting t
```

Task 2.4: Workflow Automation with GitHub Actions:

This task was also rather simple. I asked it to generate a .yml file to test my code for potential issues it may encounter if it was deployed. This was a ton easier than making the yml file myself so I thought this was a really cool and helpful feature.



```
name: Shipping Crate CI Workflow
on:
  push:
    branches:
      - main
  pull_request:
    branches:
      - main
jobs:
  build:
    runs-on: ubuntu-latest
    container: python:3.9-slim
    steps:
      - name: Checkout
        uses: actions/checkout@v3
      - name: Install dependencies
        run: |
          python -m pip install --upgrade pip
          pip install -r requirements.txt
      - name: Lint with Flake8
        run: |
          flake8 src --count --select=E9,F63,F7,F82 --show-source --statistics
          flake8 src --count --max-complexity=10 --max-line-length=127 --statistics
      - name: Run unit tests with nose
        run: nosetests -v --with-spec --spec-color --with-coverage --cover-package=src
      - name: SonarQube analysis
        run: |
          # Replace <SONAR_TOKEN> and <SONAR_PROJECT_KEY> with your SonarQube token and project key
          sonar-scanner \
            -Dsonar.projectKey=<SONAR_PROJECT_KEY> \
            -Dsonar.sources=src \
            -Dsonar.host.url=<SONARQUBE_URL> \
            -Dsonar.login=<SONAR_TOKEN>
```

Impact:

The impact of this assignment was huge. I personally have never used ChatGPT so this was kind of mind blowing to me. This software has the ability to help me understand software development better than I do now. I am able to submit any piece of code and get further clarification on how it works and how to make it better. I believe this is a great tool for learning as long as it is not abused.

Lessons Learned:

As previously mentioned I learned a ton from this. I learned how helpful this tool can be. If you are precise enough with your words you will be able to do a lot with this. I learned that wording really maters in this tool and that it is really powerful. You can make it do work for you and clear up time for other tasks (if permitted). Overall I learned that AI is not always bad and can be used for constructive purposes.

Future Applications:

After this lab I can definitely see myself using this in the future. I can see this being deployed in the work force. Unfortunately I can see a lot of jobs being replaced by AI but it may boost productivity of the human civilization. This is a little off topic but like how the industrial revolution happened, I can see the same thing happening with AI. We will be able to solve greater problems if we free up time by completing tasks with AI.

Conclusion:

Initially, I was hesitant to use ChatGPT. It has been around for a bit now but this is my first time actually using it. I have heard negative things like it being used for cheating on assignments and I wanted to stay clear from that to avoid any accusations.

After this task I was able to see that this can be a very helpful tool. As long as it is not abused and used with integrity, this can be beneficial to students and those in the workforce.

This impacted me because this displayed a new tool to use in the software development area. It shows that instead of spending time optimizing code or making comments or test files, you can make these things with AI and move on to bigger problems.

As I mentioned before, I learned a lot of lessons from this lab. It showed me a great tool I can use to boost productivity. I can also definitely see myself using this in the future. Even if I decide not to pursue a career in computer science or software development, I can use this anywhere. To improve resumes, emails, and much more. This can be used by anyone.

In conclusion, ChatGPT is a great piece of software that can be very effective in freeing up time in tasks that actually require human brain and creativity. It can be used for the simple tasks like adding comments, optimizing code, or even making test files. You can spend you time elsewhere maybe coming up with another idea. This is a great tool that has great power unless it is abused.