

Application of Maths for Software Development

by May Davies

FILE	17857_MAY_DAVIES_APPLICATION_OF_MATHS_FOR_SOFTWARE_DEV ELOPMENT_250776_1003673846.DOCX (1.55M)		
TIME SUBMITTED	30-APR-2020 08:08PM (UTC+0100)	WORD COUNT	3834
SUBMISSION ID	125729364	CHARACTER COUNT	18159

Application of Maths for Software Development

Declaration

The work of this report has been completed under the supervision of Gordan Dickers. I declare that this report is the product of my own work and is not being submitted for any other degree etc.

May Davies.

Introduction

This report is a report that has been made to help with the creation of a maths library for a new software application, in order to perform a set amount of functions including the length of a vector, the normalisation of a vector, calculating the distance between two points, dot products, cross products, rotations on y axis transformations as well as Mean, Mode, standard deviation and the union of two sets and intersection of two sets.

Before actually documenting, creating and testing the program, it will be important to describe each of these functions and how they work.

Vector length

A Vector is any object that is being moved from one location to another. It is an object travelling in a direction provided by a force [1].

To find the length of a vector, each point of the vector needs to be squared, then added up, and then the total needs to be square rooted. This is the same for both a 2D and a 3D vector.

Say for instance, $\vec{AB}$. Point A could be 27, while point B could be 12. To find the length of this vector, the following calculation would be done:

$$\sqrt{27^2 + 12^2}$$

This would output an answer of: 29.55(to 2 Decimal Places). So say this was in centimetres, the length of $\vec{AB}$ would be 29.55cm.

This length can now be referred to as $|a|$

Normalisation of a Vector

The idea behind normalisation of a vector is that a vector is divided by its length to make the number 1. A normalised vector can be represented with $\hat{a}$. So, using the previous example of $|a|=29.55$, using the following equation:

$$\hat{a} = \frac{a}{|a|}$$

Which will look like this:

$$\hat{a} = \frac{(5.196, 3.464)}{29.55}$$

Distance between 2 points

To find the distance between 2 points, you simply look at the two points, and then create a right angle triangle, and use Pythagoras theorem to get the distance ($a^2+b^2=c^2$)[3]. Though, a and b are points, so the formula would need to take this in to account.

Where x is the x axis and y is the y axis, the formula would look like this:

$$\sqrt{(Ax - Bx)^2 + (Ay - By)^2}$$

An example of this could be trying to find the distance between point A (25,3) and point B (3, 1). The results would be as follows:

$$\sqrt{(25 - 3)^2 + (3 - 1)^2} = 22.0907$$

Dot product

A dot product is a form of multiplication for 2 vectors[2]. The dot product formula can be represented with the following:

$$a \cdot b = |a||b| \cos \theta$$

If a Cos is used, the theta is the angle of the vector. Though it is possible to calculate a dot product differently, and can be shown in the form of:

$$a \cdot b = (x1 * x2 + y1 * y2 + z1 * z2)$$

Wherein:

$$a = (x1, y1, z1)$$

$$b = (x2, y2, z2)$$

So, with the following vectors:

$$a = (2, 7, 1)$$

$$b = (3, 5, 0)$$

This would become the following:

$$a \cdot b = (2 * 3 + 7 * 5 + 1 * 0)$$

Giving the dot product of:

$$a \cdot b = 41$$

Cross Product

The cross product of two vectors is effectively a third vector that can be calculated from two other vectors. This can be done by placing the two vectors in to a matrix, and then finding each 2x2 matrix within it, then finding the determinants of each matrix for the cross product[4]. So the following example assumes that $a = (2, 7, 1)$ and $b = (1, -9, 5)$.

$$\begin{array}{ccc} X & Y & Z \\ 2 & 7 & 1 \\ 1 & -9 & 5 \end{array}$$

The possible 2x2matrixes for these are as follows:

$$x \begin{array}{cc} 7 & 1 \\ -9 & 5 \end{array}$$

$$y \begin{array}{cc} 2 & 1 \\ 1 & 5 \end{array}$$

$$z \begin{array}{cc} 2 & 7 \\ 1 & -9 \end{array}$$

From here, the determinant must be found from each of these 2x2 matrices and plugged in to the formula of $x-y+z$.

The following is a formula of a determinant matrix

$$\begin{array}{cc} a & b \\ c & d \end{array} = a * d - b * c$$

So for this particular example, here would be the following determinant matrixes

$$x = (7 * 5) - (1 * -9) = 44$$

$$y = (2 * 5) - (1 * 1) = 9$$

$$z = (2 * -9) - (7 * 1) = -25$$

This means that the cross product becomes:

$$44x - 9y - 25z$$

Note: the formula would usually be $+x-y+z$, however in this instance z was a negative value.

Rotation around the y axis transformation

Due to confusion and a lack of understanding, the Y axis will unfortunately not be covered by this program.

Mean, Mode and Standard Deviation

The mean of a group of numbers is the average if all the numbers from a list are added up and then divided by the total number of items there. The following is an example of the mean:

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 = 45 \quad 45 \div 9 = 5$$

In that instance, the mean was 5.

As for the mode, the mode is the number that re-occurs the most in a series of numbers. The following is an example of a mode:

1,7,8,8,8,8,9,5,4,21,8
Mode: 8

As for standard deviation, standard deviation requires something known as variance. The variance of a set of numbers can be found by first calculating the mean, taking the mean away from each number, squaring the result of each number, and then getting the average for the squared results. This new average then needs to be square rooted, which will give the standard deviation [5].

The following is a worked example of standard deviation

$$\begin{aligned} &10, 20, 30, 25, 24, 71, 19 \\ \text{Mean} &= \frac{10 + 20 + 30 + 25 + 24 + 71 + 19}{7} = 28.4286 \text{ (to 4 d.p.)} \\ 10 - 28.4286 &= -18.4286^2 = 339.613 \\ 20 - 28.4286 &= -8.4286^2 = 71.041 \\ 30 - 28.4286 &= 1.5714^2 = 2.469 \\ 25 - 28.4286 &= -3.4286^2 = 11.755 \\ 24 - 28.4286 &= -4.4286^2 = 19.612 \\ 71 - 28.4286 &= 42.5714^2 = 1812.324 \\ 19 - 28.4286 &= -9.4286^2 = 88.898 \end{aligned}$$

$$\begin{aligned} &= 2345.712 \div 7 = 335.1017 \\ &\sqrt{335.1017} = 18.3057 \end{aligned}$$

This last number above, from the square root, is the standard deviation between results.

Union and intersection of two sets

Before defining either a union or intersection, a set is a set of items. They could be numbers, objects, names, colours, or just about anything, so long as there is at least 1 item to put in a “set”.

A union, is making a new set, that is a combination of 2 or more sets. Items do not get repeated in a union. The following is an example of a union:

$$\begin{aligned}A &= \{2,4,5,7,9,10,11\} \\B &= \{2,3,5,6,7,8,11\} \\A \cup B &= \{2,3,4,5,6,7,8,9,10,11\}\end{aligned}$$

Meanwhile, an intersection, will be a set that is only made up of any numbers that are common to all sets involved. The following is an example of an intersection:

$$\begin{aligned}A &= \{1,2,3,7,8,9,10\} \\B &= \{2,3,9,10,11,12,13\} \\A \cap B &= \{2,3,10\}\end{aligned}$$

Main Menu Printout, description, worked example and code

```
Please select which function you wish to perform from the list below:
1: Vector Length Calculator
2: Normalization of a vector
3: Distance between two points
4: Dot product of two vectors
5: Cross product of two vectors
6: Mean, Mode and Standard Deviation of a list of integers
7: Union and Intersection of two sets
Please enter the number for the option you wish to select:
```

This screenshot is a screenshot of how the main menu is presented within the program. From here, the user can choose to go to one of the 7 programmed functions of this program.

For instance, if the user enters 6, the program will call the Mean, Mode and Standard Deviation calculator. This can be shown in the screenshot below:

```
Please select which function you wish to perform from the list below:
1: Vector Length Calculator
2: Normalization of a vector
3: Distance between two points
4: Dot product of two vectors
5: Cross product of two vectors
6: Mean, Mode and Standard Deviation of a list of integers
7: Union and Intersection of two sets
Please enter the number for the option you wish to select:6

Loading Mean, Mode and Standard Deviation of a list of integers.
Please select which you wish to pick:
1: Mean calculator
2: Mode calculator
3: Standard Deviation Calculator
```

The following screenshots are of code for the main menu.

```
int main(int argc, char *argv[])
{
    //calls the main menu.
    mainMenu();
    cin.get();
    return 0;
}

void mainMenu()
{
    int menuSelect = 0;
    string input;
    stringstream stream;

    //asks the user to select which operation they wish to perform.
    while (menuSelect == 0 || menuSelect < 0 || menuSelect > 7)
    {
        cout << endl << "Please select which function you wish to perform from the list below:" << endl;
        cout << "1: Vector Length Calculator" << endl;
        cout << "2: Normalization of a vector" << endl;
        cout << "3: Distance between two points" << endl;
        cout << "4: Dot product of two vectors" << endl;
        cout << "5: Cross product of two vectors" << endl;
        cout << "6: Mean, Mode and Standard Deviation of a list of integers" << endl;
        cout << "7: Union and Intersection of two sets" << endl;
        cout << "Please enter the number for the option you wish to select:";

        getline(cin, input);
        stream.str(input);
        stream >> menuSelect;
        stream.clear();
        //calls the respective operation for the input.
        switch (menuSelect)
        {
            case 1: cout << endl << "Loading Vector Length Calculator" << endl;
                    vectorLength();
                    break;
            case 2: cout << endl << "Loading Normalization of a Vector" << endl;
                    normalizationVector();
                    break;
            case 3: cout << endl << "Loading Distance between two points" << endl;
                    twoPointDist();
                    break;
            case 4: cout << endl << "Loading Dot product of two vectors" << endl;
                    dotProd();
                    break;
            case 5: cout << endl << "Loading Cross product of two vectors" << endl;
                    crossProd();
                    break;
            case 6: cout << endl << "Loading Mean, Mode and Standard Deviation of a list of integers." << endl;
                    meanModeStandardDeviation();
                    break;
            case 7: cout << endl << "Loading Union and Intersection of two sets" << endl;
                    unionAndIntersection();
                    break;
        }
    }
}
```

```

70         break;
71         default: cout << endl << "This input is invalid. try again." << endl;
72     }
73     }
74     }
75     }
76     }
77 }

```

Vector length printout, description, worked example and code

```

Loading Vector Length Calculator
Please enter X value of the vector
20
Please enter Y value of the vector.
5
Please enter Z value of vector. If using a 2D vector, enter 0.
27
Your X is: 20
Your Y is: 5
Your Z is: 27
Vector length is: 33.9706
returning to the main menu.

```

The vector length calculators function is to allow the user to enter the X, Y and Z of a vector in order to calculate its length. In the screenshot above, the program will first ask for the x value, then the Y value, and then the Z value. Once this is done, the length is displayed to the user.

The formula to calculate a vector is as follows:

$$\sqrt{a^2 + b^2 + c^2}$$

Now, to show that the screenshot above calculates this correctly, the following calculations have been done:

$$\sqrt{20^2 + 5^2 + 27^2} = 33.9705755$$

This means that the code above is correct, as it is rounding the answer to 4 decimal places, hence 33.9706 instead of 33.9705

Vector Length Code

```

void vectorLength()
{
    double vectPointX;
    double vectPointY;
    double vectPointZ = 0;
    stringstream stream;
    //gets the user to enter a vector.
    cout << "Please enter X value of the vector" << endl;
    getline(cin, input);
    stream.str(input);
    stream >> vectPointX;
    stream.clear();
    cout << "Please enter Y value of the vector." << endl;
    getline(cin, input);
    stream.str(input);
    stream >> vectPointY;
    stream.clear();
    cout << "Please enter Z value of vector. If using a 2D vector, enter 0." << endl;
    getline(cin, input);
    stream.str(input);
    stream >> vectPointZ;
    stream.clear();
    //tells the user the vector's length.
    cout << "Your X is: " << vectPointX << endl << "Your Y is: " << vectPointY << endl << "Your Z is: " << vectPointZ << endl;
    cout << "Vector length is: " << sqrt((vectPointX * vectPointX) + (vectPointY * vectPointY) + (vectPointZ * vectPointZ)) << endl;
    cout << "returning to the main menu." << endl;
    mainMenu();
}

```

Normalization of vector printout, description, worked example and code

```
Loading Normalization of a Vector
This is the normalization of a vector calculator
Please enter the X of your vector:5

Please enter the Y of your vector:2

Please enter the Z of your vector:6

Please enter the length of your vector:8.06226

Your normalized vector is: (0.620173,0.248069,0.744208)
Returning to main menu.
```

Normalization of a vector is when a vectors points are each divided by the length of the vector, which outputs a normalized vector as an answer.

This can be shown as:

$$\hat{a} = \frac{a}{|a|}$$

The following workings have been done to check that the code for this section is correct:

$$\left(\frac{5}{8.06226}\right) = 0.620173$$

$$\left(\frac{2}{8.06226}\right) = 0.248069$$

$$\left(\frac{6}{8.06226}\right) = 0.744208$$

The workings match the result shown in the screenshot of the program.

Normalization vector code

```
void normalizationVector()
{
    double vectPointX;
    double vectPointY;
    double vectPointZ = 0;
    double vectLength;
    string input;
    stringstream stream;

    //asks the user for a vector and its length.
    cout << "This is the normalization of a vector calculator" << endl;
    cout << "Please enter the X of your vector:";
    getline(cin, input);
    stream.str(input);
    stream >> vectPointX;
    stream.clear();
    cout << endl << "Please enter the Y of your vector:";
    getline(cin, input);
    stream.str(input);
    stream >> vectPointY;
    stream.clear();
    cout << endl << "Please enter the Z of your vector:";
    getline(cin, input);
    stream.str(input);
    stream >> vectPointZ;
    stream.clear();
    cout << endl << "Please enter the length of your vector:";
    getline(cin, input);
    stream.str(input);
    stream >> vectLength;
    stream.clear();
    //tells the user the normalized vector from the length and vector input.
    cout << endl << "Your normalized vector is: (" << vectPointX / vectLength << ", " << vectPointY / vectLength << ", " << vectPointZ / vectLength << ")" << endl;
    cout << "Returning to main menu." << endl;
    mainMenu();
}
```

Distance between 2 points printout, description, worked example and code

```
Loading Distance between two points
This is the distance between 2 points calculator.
Please enter the X value of point A:25

Please enter the Y value of point A:3
Please enter the X value of point B:3

Please enter the Y value of point B:1

The distance between point A(25,1) is 22.0907
Returning to the main menu.
```

The distance between 2 points calculator of the program calculates the distance between two points by taking the firstly taking the x of B away from the x of A, squaring that answer, and then taking the y of B away from the y of a, squaring that answer, adding the two answers and then square rooting them to find the distance, which can be represented with the following formula:

$$\sqrt{(Ax - Bx)^2 + (Ay - By)^2}$$

The following screenshot can be proven by using the workings below:

$$\sqrt{(25 - 3)^2 + (3 - 1)^2} = 22.0907$$

Distance between 2 points code

```
void twoPointDist()
{
    double pointAX;
    double pointAY;
    double pointBX;
    double pointBY;
    string input;
    stringstream stream;

    cout << "This is the distance between 2 points calculator." << endl;
    // asks the user to enter two points
    cout << "Please enter the X value of point A:";
    getline(cin, input);
    stream.str(input);
    stream >> pointAX;
    stream.clear();
    cout << endl << "Please enter the Y value of point A:";
    getline(cin, input);
    stream.str(input);
    stream >> pointAY;
    stream.clear();
    cout << "Please enter the X value of point B:";
    getline(cin, input);
    stream.str(input);
    stream >> pointBX;
    stream.clear();
    cout << endl << "Please enter the Y value of point B:";
    getline(cin, input);
    stream.str(input);
    stream >> pointBY;
    stream.clear();
    //outputs the calculated distance between the two points in a message to the user.
    cout << endl << "The distance between point A(" << pointAX << ", " << pointAY << ") is " << sqrt(((pointAX-pointBX)*(pointAX - pointBX))+ ((pointAY-pointBY)*(pointAY - pointBY))) << endl;
    cout << "Returning to the main menu." << endl;
    mainMenu();
}
```

Dot product printout, description, workings and code

```
Loading Dot product of two vectors
Please select how you would like the dot product calculated:
1: using cosine theta
2: without using cosine theta
1
Using cosine dot product calculation
Please enter length A:3.4641
Please enter length B:12.1244
Please enter the value for theta:30
a.b = 6.47858 radians.
Returning to the main menu.
```

```
Loading Dot product of two vectors
Please select how you would like the dot product calculated:
1: using cosine theta
2: without using cosine theta
2
Not using cosine dot product calculation
Please enter X for point A:2
Please enter Y for point A:7
Please enter Z for point A:1
Please enter X for point B:3
Please enter Y for point B:5
Please enter Z for point B:0
a.b = 41
Returning to main menu.
```

This program can calculate a dot product in two different ways. The first way being by multiplying two lengths together and then cosining that length by an angle, while the second is by multiplying each dimension by their respective dimension in the other vector, and then adding the results.

The cosine option outputs in radians, and the formula for it can be seen below:

$$a.b = |a||b| \cos \theta$$

In this instance, the workings are as follows:

$$3.4641 * 12.1244 * \cos 30 = 6.47858$$

Meanwhile, the formula for finding the dot product without a cosine can be seen here:

$$a.b = (x1 * x2 + y1 * y2 + z1 * z2)$$

The following calculation can determine that this version of the dot product is calculated correctly to prove the second screenshot as seen above:

$$a = (2, 7, 1)$$

$$b = (3, 5, 0)$$

$$a.b = (2 * 3 + 7 * 5 + 1 * 0) = 41$$

Dot Product Code

```
void dotProd()
{
    double lengthA;
    double lengthB;
    double thetaVal;
    double pointAX;
    double pointAY;
    double pointAZ;
    double pointBX;
    double pointBY;
    double pointBZ;
    int menuSelect = 0;
    string input;
    stringstream stream;

    //Asks the user to pick between either calculating by using cosine or not using cosine.
    while (menuSelect == 0 || menuSelect < 0 || menuSelect > 2)
    {
        cout << "Please select how you would like the dot product calculated:" << endl;
        cout << "1: using cosine theta" << endl;
        cout << "2: without using cosine theta" << endl;
        getline(cin, input);
        stream.str(input);
        stream >> menuSelect;
        stream.clear();
        switch (menuSelect)
        {
            case 1: cout << endl << "Using cosine dot product calculation" << endl;
                    //asks the user to enter two vector lengths.
                    cout << "Please enter length A:";
                    getline(cin, input);
                    stream.str(input);
                    stream >> lengthA;
                    stream.clear();
                    cout << endl;
                    cout << "Please enter length B:";
                    getline(cin, input);
                    stream.str(input);
                    stream >> lengthB;
                    stream.clear();
                    cout << endl;
        }
```

```

    cout << "Please enter the value for theta:";
    getline(cin, input);
    stream.str(input);
    stream >> thetaVal;
    stream.clear();
    cout << endl;
    //uses the length of vector A and the length of vector B multiplied by a Cosined
    cout << "a.b = " << (lengthA * lengthB * cos(thetaVal)) << " radiuns." << endl;
    cout << "Returning to the main menu." << endl;
    mainMenu();
    break;
case 2: cout << endl << "Not using cosine dot product calculation" << endl;
    //asks for 2 vectors.
    cout << "Please enter X for point A:";
    getline(cin, input);
    stream.str(input);
    stream >> pointAX;
    stream.clear();
    cout << endl;
    cout << "Please enter Y for point A:";
    getline(cin, input);
    stream.str(input);
    stream >> pointAY;
    stream.clear();
    cout << endl;
    cout << "Please enter Z for point A:";
    getline(cin, input);
    stream.str(input);
    stream >> pointAZ;
    stream.clear();
    cout << endl;

    cout << "Please enter X for point B:";
    getline(cin, input);
    stream.str(input);
    stream >> pointBX;
    stream.clear();
    cout << endl;
    cout << "Please enter Y for point B:";

    getline(cin, input);
    stream.str(input);
    stream >> pointBY;
    stream.clear();
    cout << endl;
    cout << "Please enter Z for point B:";
    getline(cin, input);
    stream.str(input);
    stream >> pointBZ;
    stream.clear();
    cout << endl;
    // calculates the dot product with the formula of (Ax*Bx)+(Ay*By)+(Az*Bz).
    cout << "a.b = " << ((pointAX * pointBX) + (pointAY * pointBY) + (pointAZ * pointBZ)) << endl;
    cout << "Returning to main menu." << endl;
    mainMenu();
    break;
default: cout << "invalid input try again." << endl;
}
}
}

```

Cross Product printout, description, workings and code

```
Loading Cross product of two vectors

Please enter Vector 0: (2
,7
,1
)

Please enter Vector 1: (1
,-9
,5
)

This is the X matrix:
7  1
-9  5

This is the Y matrix:
2  1
1  5

This is the Z matrix:
2  7
1 -9

The cross product is: (44,9,-25)
Returning to main menu.
```

A cross product is the result of finding the determinants of several 2x2 matrixes made up of 2 vectors. The cross product is essentially a 3rd vector created from 2 already existing vectors.

The formula for a cross product is as follows:

$$\begin{array}{ccc} X & Y & Z \\ x1 & y1 & z1 \\ x2 & y2 & z2 \end{array}$$

$$\begin{array}{cc} x & \begin{array}{cc} y1 & z1 \\ y2 & z2 \end{array} \end{array}$$

$$\begin{array}{cc} y & \begin{array}{cc} x1 & z1 \\ x2 & z2 \end{array} \end{array}$$

$$\begin{array}{cc} z & \begin{array}{cc} x1 & y1 \\ x2 & y2 \end{array} \end{array}$$

$$\text{Determinant cross product} = \begin{array}{cc} a & b \\ c & d \end{array} = a * d - b * c$$

In this instance, the workings to prove the screenshot example can be seen below:

$$\begin{array}{ccc} X & Y & Z \\ 2 & 7 & 1 \\ 1 & -9 & 5 \end{array}$$

$$\begin{array}{cc} x & \begin{array}{cc} 7 & 1 \\ -9 & 5 \end{array} \end{array}$$

$$\begin{array}{cc} y & \begin{array}{cc} 2 & 1 \\ 1 & 5 \end{array} \end{array}$$

$$\begin{array}{cc} z & \begin{array}{cc} 2 & 7 \\ 1 & -9 \end{array} \end{array}$$

$$x = (7 * 5) - (1 * -9) = 44$$

$$y = (2 * 5) - (1 * 1) = 9$$

$$z = (2 * -9) - (7 * 1) = -25$$

$$\text{Cross product} = (44x - 9y - 25z)$$

Cross product code

```

282 void crossProd()
283 {
284     double vectMatrix[2][3];
285     double xMatrix[2][2];
286     double yMatrix[2][2];
287     double zMatrix[2][2];
288     double xDeterminant;
289     double yDeterminant;
290     double zDeterminant;
291     int count = 0;
292     int count2 = 0;
293     string input;
294     stringstream stream;
295     for (count = 0; count < 2; count++)
296     {
297         //asks the user for a vector, then stores the vector as a m
298         cout << endl << "Please enter Vector "<<count<<": ";
299         getline(cin, input);
300         stream.str(input);
301         stream >> vectMatrix[count][0];
302         stream.clear();
303         cout << ", ";
304         getline(cin, input);
305         stream.str(input);
306         stream >> vectMatrix[count][1];
307         stream.clear();
308         cout << ", ";
309         getline(cin, input);
310         stream.str(input);
311         stream >> vectMatrix[count][2];
312         stream.clear();
313         cout << ")" << endl;
314     }
315     count = 0;
316     // 2x2 matrixes are made to be used to calculate Determinants.
317     xMatrix[0][0] = vectMatrix[0][1];
318     xMatrix[0][1] = vectMatrix[0][2];
319     xMatrix[1][0] = vectMatrix[1][1];

```

```

320     xMatrix[1][1] = vectMatrix[1][2];
321
322     yMatrix[0][0] = vectMatrix[0][0];
323     yMatrix[0][1] = vectMatrix[0][2];
324     yMatrix[1][0] = vectMatrix[1][0];
325     yMatrix[1][1] = vectMatrix[1][2];
326
327     zMatrix[0][0] = vectMatrix[0][0];
328     zMatrix[0][1] = vectMatrix[0][1];
329     zMatrix[1][0] = vectMatrix[1][0];
330     zMatrix[1][1] = vectMatrix[1][1];
331     // displays each matrix to the user.
332     cout << "This is the X matrix:" << endl;
333     for (count = 0; count < 2; count++)
334     {
335         for (count2 = 0; count2 < 2; count2++)
336         {
337             cout << xMatrix[count][count2] << " ";
338         }
339         cout << endl;
340     }
341     count = 0;
342     count2 = 0;
343     cout << "This is the Y matrix:" << endl;
344     for (count = 0; count < 2; count++)
345     {
346         for (count2 = 0; count2 < 2; count2++)
347         {
348             cout << yMatrix[count][count2] << " ";
349         }
350         cout << endl;
351     }
352     count = 0;
353     count2 = 0;
354     cout << "This is the Z matrix:" << endl;
355     for (count = 0; count < 2; count++)

```

```

358 {
359     for (count2 = 0; count2 < 2; count2++)
360     {
361         cout << zMatrix[count][count2] << " ";
362     }
363     cout << endl;
364 }
365
366 count = 0;
367 count2 = 0;
368 //calculates the determinant respective of each dimension of the new vector that is the Cross Product.
369 xDeterminant = (xMatrix[0][0] * xMatrix[1][1]) - (xMatrix[0][1] * xMatrix[1][0]);
370 yDeterminant = (yMatrix[0][0] * yMatrix[1][1]) - (yMatrix[0][1] * yMatrix[1][0]);
371 zDeterminant = (zMatrix[0][0] * zMatrix[1][1]) - (zMatrix[0][1] * zMatrix[1][0]);
372
373 cout << endl;
374 //Displays the cross product to the user.
375 cout << "The cross product is: (" << xDeterminant << ", " << yDeterminant << ", " << zDeterminant << ")" << endl;
376 cout << "Returning to main menu." << endl;
377 mainMenu();
378
379 }

```

Mean, Mode and Standard deviation printout, description, workings and code

```
Loading Mean, Mode and Standard Deviation of a list of integers.
Please select which you wish to pick:
1: Mean calculator
2: Mode calculator
3: Standard Deviation Calculator
1

Mean Calculator
please enter how many numbers will be entered up to 100:5
Please enter input number 0:10
Please enter input number 1:10
Please enter input number 2:10
Please enter input number 3:10
Please enter input number 4:10
The mean of your values is:10
Returning to the main menu.
```

```
Mode Calculator
Please enter the range of numbers you will enter:4
Please enter input number 0:1
Please enter input number 1:2
Please enter input number 2:1
Please enter input number 3:3
Your sorted numbers are:
1
1
2
3
Mode is: 1
Returning to main menu.
```

```
Standard Deviation calculator
please enter how many numbers will be entered up to 100:7
Please enter input number 0:10
Please enter input number 1:20
Please enter input number 2:30
Please enter input number 3:25
Please enter input number 4:24
Please enter input number 5:71
Please enter input number 6:19
The mean of your values is:28.4286
Variance for number: 10 is 339.612
Variance for number: 20 is 71.0408
Variance for number: 30 is 2.46939
Variance for number: 25 is 11.7551
Variance for number: 24 is 19.6122
Variance for number: 71 is 1812.33
Variance for number: 19 is 88.898
Standard Deviation is: 48.4326
Returning to the main menu.
```

This part of the program is a part of the program that can calculate either the mean, mode or standard deviation of a group of numbers.

The first part of the programs formula (the mean) looks like this:

$$mean = \frac{values}{NumberOfValues}$$

The workings for the mean are listed below to prove the mean works as intended:

$$mean = \frac{10 + 10 + 10 + 10 + 10}{5} = 10$$

The second part of this part of the program is the mode. The mode is the number that appears the most often in a range of numbers.

In this case, the following workings can be done to prove the 2nd screenshot:

Inputs: 1, 1, 2, 3

Mode: 1 as it appears twice, all other numbers appear less

The final part of this part of the program is standard deviation. Standard deviation involves initially finding the mean of a range of numbers, then taking away the mean from each input to find the variation. The variation then gets added up, and then a new mean is calculated from the variation, and then that number is square rooted to output the standard deviance.

The following workings can represent the screenshot of the program:

$$\text{Mean} = \frac{10 + 20 + 30 + 25 + 24 + 71 + 19}{7} = 28.4286 \text{ (to 4 d.p)}$$

$$10 - 28.4286 = -18.4286^2 = 339.613$$

$$20 - 28.4286 = -8.4286^2 = 71.041$$

$$30 - 28.4286 = 1.5714^2 = 2.469$$

$$25 - 28.4286 = -3.4286^2 = 11.755$$

$$24 - 28.4286 = -4.4286^2 = 19.612$$

$$71 - 28.4286 = 42.5714^2 = 1812.324$$

$$19 - 28.4286 = -9.4286^2 = 88.898$$

$$= 2345.712 \div 7 = 335.1017$$

$$\sqrt{335.1017} = 18.3057$$

It should be worth noting that some of these answers in these workings wont match up to the screenshot due to inaccuracies caused by rounding up or down to a number of decimal places.

Code for Mean, Mode and Standard Deviation

```

380 void meanModeStandardDeviation()
381 {
382     int menuSelect = 0;
383     int numRange = 0;
384     int sortRange = 0;
385     double inputForCalc[100];
386     string input;
387     stringstream stream;
388     int count = 0;
389     int count2 = 0;
390     double total = 0;
391     double temp1;
392     double modeCount[100][2];
393     double maxVal = -999999;
394     double mode = -999999;
395     double sdMean;
396     double variance[100];
397     double varianceTotal = 0;
398
399
400     while (menuSelect == 0 || menuSelect < 0 || menuSelect > 3)
401     {
402         cout << "Please select which you wish to pick:" << endl;
403         cout << "1: Mean calculator" << endl;
404         cout << "2: Mode calculator" << endl;
405         cout << "3: Standard Deviation Calculator" << endl;
406
407         getline(cin, input);
408         stream.str(input);
409         stream >> menuSelect;
410         stream.clear();
411         //Asks the user to select if they want to calculate the mean, mode or st
412         switch (menuSelect)
413         {
414             case 1: cout << endl << "Mean Calculator" << endl;
415                     cout << "please enter how many numbers will be entered up to 100:";
416                     getline(cin, input);
417                     stream.str(input);
418

```

```

419 stream >> numRange;
420 stream.clear();
421 for (count = 0; count < numRange; count++)
422 {
423     cout << endl << "Please enter input number " << count << " ";
424     getline(cin, input);
425     stream.str(input);
426     stream >> inputForCalc[count];
427     stream.clear();
428     //stores the total for later.
429     total = total + inputForCalc[count];
430 }
431 //displays the average by dividing by the total.
432 cout << "The mean of your values is!" << total / numRange << endl;
433 cout << "Returning to the main menu." << endl;
434 mainMenu();
435 break;
436 case 2: cout << endl << "Mode Calculator" << endl;
437 //allows the user to enter a range of numbers in order to calculate t
438 cout << "Please enter the range of numbers you will enter:";
439 getline(cin, input);
440 stream.str(input);
441 stream >> numRange;
442 stream.clear();
443 //Sorting algorithm was used since there didnt seem to be a sort() fu
444 //more research is needed on this for any possible sort libraries.
445 sortRange = numRange - 1;
446 for (count = 0; count < numRange; count++)
447 {
448     cout << endl << "Please enter input number " << count << " ";
449     getline(cin, input);
450     stream.str(input);
451     stream >> inputForCalc[count];
452     stream.clear();
453 }
454 for (count = 0; count < numRange; count++)
455 {
456     for (count2 = 0; count2 < sortRange; count2++)
457     {
458         if (inputForCalc[count2] > inputForCalc[count2 + 1])
459         {
460             //Swaps the numbers around in order to sort them, effecti
461             temp1 = inputForCalc[count2];
462             inputForCalc[count2] = inputForCalc[count2 + 1];
463             inputForCalc[count2 + 1] = temp1;
464         }
465     }
466 }

```

```

468 }
469 //This is just here purely to show the numbers are indeed sorted.
470 cout << "Your sorted numbers are: " << endl;
471 for (count = 0; count < numRange; count++)
472 {
473     cout << inputForCalc[count] << endl;
474     modeCount[count][0] = inputForCalc[count];
475     modeCount[count][1] = 0;
476 }
477 for (count = 0; count < numRange; count++)
478 {
479     for (count2 = 0; count2 < numRange; count2++)
480     {
481         if (inputForCalc[count2] == modeCount[count][0])
482         {
483             modeCount[count][1]++;
484             if (modeCount[count][1] > maxVal)
485             {
486                 maxVal = modeCount[count][1];
487                 mode = modeCount[count][0];
488             }
489         }
490     }
491     //cout << "Number " << modeCount[count][0] << " came up " << m
492 }
493 //This code is so spaghetti even an Italian would be scared to eat
494 //Displays the mode to the user.
495 cout << "Mode is: " << mode << endl;
496 cout << "Returning to main menu." << endl;
497 mainMenu();
498 break;
499 case 3: cout << endl << "Standard Deviation calculator" << endl;
500 //asks the user for a range of numbers in order to get an initial a
501 cout << "Please enter how many numbers will be entered up to 100:";
502 getline(cin, input);
503 stream.str(input);
504 stream >> numRange;
505 stream.clear();
506 for (count = 0; count < numRange; count++)
507 {
508     //user inputs numbers for mean calculation.
509     cout << endl << "Please enter input number " << count << " ";
510     getline(cin, input);
511     stream.str(input);
512     stream >> inputForCalc[count];
513     stream.clear();
514     total = total + inputForCalc[count];
515 }
516 }

```

```

517 //displays the mean to the user.
518 cout << "The mean of your values is:" << total / numRange << endl;
519 sdMean = total / numRange;
520 //Calculates the variance and then stores in a variance array.
521 for (count = 0; count < numRange; count++)
522 {
523     variance[count] = (inputForCalc[count] - sdMean);
524     variance[count] = variance[count] * variance[count];
525     cout << "Variance for number: " << inputForCalc[count] << " is " << variance[count] << endl;
526     //gets the total variance.
527     varianceTotal = varianceTotal + variance[count];
528 }
529 //standard deviation is calculated by square rooting the total variance and then outputting to the user.
530 cout << "Standard Deviation is: " << sqrt(varianceTotal) << endl;
531
532
533
534
535
536     cout << "Returning to the main menu.";
537     mainMenu();
538     break;
539 default: cout << "invalid input try again." << endl;
540 }
541 }
542 }
543 }

```

Union and Intersection of two sets printout, description, workings and code

```
Loading Union and Intersection of two sets
How long is Set A?:3
How long is Set B?:2
Please enter set A item number 0:1
Please enter set A item number 1:2
Please enter set A item number 2:3
Please enter set B item number 0:4
Please enter set B item number 1:5
The intersect is as follows: {}
The union is as follows: {1,2,3,4,5,}
Returning to main menu.

Please enter set A item number 0:1
Please enter set A item number 1:2
Please enter set A item number 2:3
Please enter set B item number 0:1
Please enter set B item number 1:2
The intersect is as follows: {1,2,}
The union is as follows: {3,}
Returning to main menu.
```

The union and intersection of a set part of the program allows the user to see which items in a set appear in both (the intersect), and how the sets look when combined (Union).

The following formula can represent 2 sets, a union and an intersect:

$$\text{set } a = \{[\text{Item } 1], [\text{Item } 2], [\text{Item } 3] \dots\}$$

$$\text{set } b = \{[\text{item } 2], [\text{item } 7], [\text{item } 9], [\text{item } 1] \dots\}$$

$$a \cup b = \{[\text{item } 1], [\text{item } 2], [\text{item } 3], [\text{item } 7], [\text{item } 9]\}$$

$$a \cap b = \{[\text{item } 1], [\text{item } 2]\}$$

Already from the formula alone, looking at the screenshots, something is wrong with the code, this has yet to be worked around. The following workings depict the results that the code should have shown:

Union:

$$\text{set } a = \{1,2,3\}$$

$$\text{set } b = \{4,5\}$$

$$a \cup b = \{1,2,3,4,5\}$$

$$a \cap b = \{\}$$

Intersect:

$$\text{set } a = \{1,2,3\}$$

$$\text{set } b = \{1,2\}$$

$$a \cup b = \{1,2,3\}$$

$$a \cap b = \{1,2\}$$

Union and intersect code

```
544 void unionAndIntersection()
545 {
546     string setA[100];
547     int setALength;
548     string setB[100];
549     int setBLength;
550     string unionSet[200];
551     int unionLength;
552     string intersectionSet[100];
553     int count = 0;
554     int count2 = 0;
555     int intersectionSetLength;
556     int intersectCounter = 0;
557     int unionCounter = 0;
558     string input;
559     stringstream stream;
560     //asks for a set length.
561     cout << endl << "How long is Set A?:";
562     getline(cin, input);
563     stream.str(input);
564     stream >> setALength;
565     stream.clear();
566     //asks for b set length.
567     cout << endl << "How long is Set B?:";
568     getline(cin, input);
569     stream.str(input);
570     stream >> setBLength;
571     stream.clear();
572
573     //calculates the maximum possible length for a union of the two sets.
574     unionLength = setALength + setBLength;
575
576     for (count = 0; count < setALength; count++)
577     {
578         //allows input of items for set A.
579         cout << endl << "Please enter set A item number " << count << ":";
580         getline(cin, setA[count]);
581     }
582     for (count = 0; count < setBLength; count++)
583     {
584         //allows input of items for set B.
585         cout << endl << "Please enter set B item number " << count << ":";
586         getline(cin, setB[count]);
587     }
588 }
```

```
589 //checks which set is longer in order to use as a base length
590 if (setALength > setBLength)
591 {
592     intersectionSetLength = setALength;
593 }
594 else
595 {
596     intersectionSetLength = setBLength;
597 }
598
599 //loop for finding the correct intersect.
600 for(count = 0; count < intersectionSetLength; count++)
601 {
602     for (count2 = 0; count2 < intersectionSetLength; count2++)
603     {
604         if (setA[count] == setB[count2])
605         {
606             intersectionSet[intersectCounter] = setB[count2];
607             intersectCounter++;
608         }
609     }
610 }
611
612 //Displays intersect to the user.
613 cout << endl << "The intersect is as follows: {";
614 for (count = 0; count < intersectCounter; count++)
615 {
616
617     cout << intersectionSet[count] << ",";
618
619 }
620 cout << "}" << endl;
621
622
623 //This code should have output the union, though it does not work
624 cout << endl << "The union is as follows: {";
625 for (count = 0; count < unionLength; count++)
626 {
627     /// for (count2 = 0; count2 < unionLength; count2++)
628     /// {
629     if ((setA[count] != setB[count]) && setA[count] != "")
630     {
631         unionSet[unionCounter] = setA[count];
632         unionCounter++;
633     }
634 }
635 }
```

```

636     /// }
637 }
638 for (count = 0; count < unionLength; count++)
639 {
640     /// for (count2 = 0; count2 < unionLength; count2++)
641     /// {
642         if ((setB[count] != setA[count]) && setB[count] != "")
643         {
644             unionSet[unionCounter] = setB[count];
645             unionCounter++;
646         }
647     }
648     /// }
649 }
650
651
652
653
654 //This code would have displayed the union correctly to the use
655 //The displayed union is not always correct.
656 for (count = 0; count < unionCounter; count++)
657 {
658
659
660
661     cout << unionSet[count] << ",";
662
663 }
664 cout << "}" << endl;
665 cout << "Returning to main menu." << endl;
666 mainMenu();
667
668
669 }

```

Testing

This is the final part of documentation. This part of the documentation revolves around testing the entirety of this program to ensure that it works as intended. Testing will be split in to sections based on different functions of the program.

Testing Main Menu

Test Number	Desc.	Expected result	Actual result	Success?
1	Check if user can navigate to Vector calculator	Moves to vector calculator	Please enter the number for the option you wish to select:1 Loading Vector Length Calculator	Yes
2	Check if user can navigate to Normalization	Moves to normalization	Please enter the number for the option you wish to select:2 Loading Normalization of a Vector	Yes
3	Check if user can navigate to distance between 2 points	Moves to distance between 2 points	Please enter the number for the option you wish to select:3 Loading Distance between two points	Yes
4	Check if user can navigate to Dot Product Calculator	Moves to Dot Product Calculator	Please enter the number for the option you wish to select:4 Loading Dot product of two vectors	Yes
5	Check if user can navigate to Cross Product Calculator	Moves to Cross Product Calculator	Please enter the number for the option you wish to select:5 Loading Cross product of two vectors	Yes
6	Check if user can navigate to Mean Mode Standard Deviation	Moves to Mean Mode Standard Deviation	Please enter the number for the option you wish to select:6 Loading Mean, Mode and Standard Deviation of a list of integers.	Yes
7	Check if user can navigate to union and intersection	Moves to union and intersection	Please enter the number for the option you wish to select:7 Loading Union and Intersection of two sets	Yes
8	Enter an invalid input for navigation	Makes the user re-enter menu selection option	Please enter the number for the option you wish to select:-9 This input is invalid. try again. Please enter the number for the option you wish to select:bob This input is invalid. try again. Please enter the number for the option you wish to select:20 This input is invalid. try again.	Yes

Vector length testing

Test	Desc.	Expected result	Actual Result	Success?
1	Can navigate from main menu to vector length	Load vector length calculator	Please enter the number for the option you wish to select:1 Loading Vector Length Calculator	Yes
2	Can calculate a vector of (20, 5, 7)	33.9706	Vector length is: 33.9706 returning to the main menu	Yes
3	Check to see if a vector of (0,0,0) outputs either 0 or NaN instead of crashing	Output either 0 or NaN without crashing	Vector length is: 0	Yes
4	Check to see if a vector of (bob, jeff, jim) gets input in to x, y and z and if it gets output as 0 or NaN	X=0 or NaN Y=0 or NaN Z=0 or NaN Length = 0 or NaN	Your X is: 0 Your Y is: 0 Your Z is: 0 Vector length is: 0	Yes
5	Check to see that the program returns to the main menu once the length has been calculated	Should return to the main menu.	returning to the main menu. Please select which function you wish to perform from the list below: 1: Vector Length Calculator 2: Normalization of a vector 3: Distance between two points 4: Dot product of two vectors 5: Cross product of two vectors 6: Mean, Mode and Standard Deviation of a list of integers 7: Union and Intersection of two sets Please enter the number for the option you wish to select:	Yes

Normalisation of a Vector testing

Test	Desc.	Expected Result	Actual Result	Success?
1	Can navigate from the main menu to normalization	Should move from the main menu to normalization	Please enter the number for the option you wish to select:2 Loading Normalization of a Vector	Yes
2	Enter a vector of (5, 2, 6) with a length of 8.06226	Output a normalized vector of (0.620173, 0.248069, 0.744208)	This is the normalization of a vector calculator Please enter the X of your vector:5 Please enter the Y of your vector:2 Please enter the Z of your vector:6 Please enter the length of your vector:8.06226 Your normalized vector is: (0.620173,0.248069,0.744208)	Yes
3	Entering 0 for every value	Output for normalized vector should either be NaN or 0 without crashing.	Please enter the X of your vector:0 Please enter the Y of your vector:0 Please enter the Z of your vector:0 Please enter the length of your vector:0 Your normalized vector is: (-nan(ind),-nan(ind),-nan(ind))	
4	Entering "bob", 2, "Yes" for vector, with a length of "6B"	Should output either 0 or nan for normalized vector without crashing	Please enter the X of your vector:bob Please enter the Y of your vector:2 Please enter the Z of your vector:yes Please enter the length of your vector:6B Your normalized vector is: (0,0.333333,0)	Mostly. Appears to read 6B as an actual number, so Y, which is 2, gives a normalized vector of 0.33 recurring.
5	Entering nothing for vector and length	Output should be either NaN, or inf.	Please enter the X of your vector: Please enter the Y of your vector: Please enter the Z of your vector: Please enter the length of your vector: Your normalized vector is: (1,1,-0)	No. Interesting that it gives the final part of the normalized vector as negative 0 though. For future versions a fix for this

				would be to prevent the user from leaving an input blank, or setting the default values to 0.
6	Check to see that the user can go back to the main menu once the normalization has finished	User should be directed back to the main menu.	<pre> Your normalized vector is: (1,1,-0) Returning to main menu. Please select which function you wish to perform from the list below: 1: Vector Length Calculator 2: Normalization of a vector 3: Distance between two points 4: Dot product of two vectors 5: Cross product of two vectors 6: Mean, Mode and Standard Deviation of a list of integers 7: Union and Intersection of two sets Please enter the number for the option you wish to select: </pre>	

Distance between 2 points tests

Test	Desc.	Expected result	Actual result	Success?
1	Check that the user can navigate to the distance calculator from the main menu	Navigates from the main menu to the distance between 2 points calculator	<pre> 7. Union and Intersection of two sets Please enter the number for the option you wish to select:3 Loading Distance between two points This is the distance between 2 points calculator. Please enter the X value of point A: </pre>	Yes
2	Input a point with the co-ordinates of (25, 3) and a point with the co-ordinates of (3, 1)	Expected distance should be 22.0907	<pre> Please enter the X value of point A:25 Please enter the Y value of point A:3 Please enter the X value of point B:3 Please enter the Y value of point B:1 The distance between point A(25,1) is 22.0907 </pre>	Yes
3	Check to see what happens when all values are entered as 0	Output should be 0.	<pre> This is the distance between 2 points calculator Please enter the X value of point A:0 Please enter the Y value of point A:0 Please enter the X value of point B:0 Please enter the Y value of point B:0 The distance between point A(0,0) is 0 </pre>	Yes
4	Check to see what happens when outputs are entered as "Jack", "FL", 5, "true"	Output will either be 0 due to values being invalid, or will be 5 due to every other value being set to 0 while one point has a co-ordinate of 5.	<pre> Please enter the X value of point A:Jack Please enter the Y value of point A:FL Please enter the X value of point B:5 Please enter the Y value of point B:true The distance between point A(0,0) is 5 </pre>	Yes. 5 is correct here as it shows that the values for point A are set to 0. Also noticed that there is not an output message for point B here,

				however this does not affect the answer that will be given to the user.
5	Check what happens when all values are "Bob"	Output should be 0 or nan.	<pre> This is the distance between 2 points calculator. Please enter the X value of point A:bob Please enter the Y value of point A:bob Please enter the X value of point B:bob Please enter the Y value of point B:bob The distance between point A(0,0) is 0 </pre>	Yes
6	Check to see that the user gets redirected back to the main menu once the distance has been calculated	User should return to the main menu	<pre> The distance between point A(0,0) is 0 Returning to the main menu. Please select which function you wish to perform from the list below: 1: Vector Length Calculator 2: Normalization of a vector 3: Distance between two points 4: Dot product of two vectors 5: Cross product of two vectors 6: Mean, Mode and Standard Deviation of a list of integers 7: Union and Intersection of two sets Please enter the number for the option you wish to select: </pre>	Yes

Dot Product testing

test	Desc.	Expected result	Actual result	Success?
1	Check to see the user can navigate to the Dot Product Menu from the main menu	User should be able to navigate to the dot product menu from the main menu	<pre> Please select which function you wish to perform from the list below: 1: Vector Length Calculator 2: Normalization of a vector 3: Distance between two points 4: Dot product of two vectors 5: Cross product of two vectors 6: Mean, Mode and Standard Deviation of a list of integers 7: Union and Intersection of two sets Please enter the number for the option you wish to select:4 Loading Dot product of two vectors Please select how you would like the dot product calculated: 1: using cosine theta 2: without using cosine theta </pre>	Yes
2	Input 0 for dot product calculation selection	User should be asked again	<pre> Loading Dot product of two vectors Please select how you would like the dot product calculated: 1: using cosine theta 2: without using cosine theta 0 invalid input try again. Please select how you would like the dot product calculated: 1: using cosine theta 2: without using cosine theta </pre>	Yes
3	Input 3 for dot product calculation selection	User should be asked again	<pre> Please select how you would like the dot product calculated: 1: using cosine theta 2: without using cosine theta 3 invalid input try again. Please select how you would like the dot product calculated: 1: using cosine theta 2: without using cosine theta </pre>	Yes
4	Input "bob" for dot product calculation selection	User should be asked again	<pre> Please select how you would like the dot product calculated: 1: using cosine theta 2: without using cosine theta bob invalid input try again. Please select how you would like the dot product calculated: 1: using cosine theta 2: without using cosine theta </pre>	Yes
5	Input 1, in order to load the cosine theta calculation	User will be asked to input values needed for the cosine theta calculation of the dot product	<pre> Please select how you would like the dot product calculated: 1: using cosine theta 2: without using cosine theta 1 Using cosine dot product calculation Please enter length A: </pre>	Yes

6	Input 3.4641 for length a, 12.1244 for length b and 30 for theta.	Output should be 6.47858	Using cosine dot product calculation Please enter length A:3.4641 Please enter length B:12.1244 Please enter the value for theta:30 a.b = 6.47858 radiuns.	Yes
7	Input "bob" for length a, "Jeff" for length b and "Jane" for theta.	Outputs should be 0 or NaN	Using cosine dot product calculation Please enter length A:bob Please enter length B:Jeff Please enter the value for theta:Jane a.b = 0 radiuns.	Yes
8	Input 0 for length a, 0 for length b and 0 for theta	Output should be 0	Using cosine dot product calculation Please enter length A:0 Please enter length B:0 Please enter the value for theta:0 a.b = 0 radiuns.	Yes
9	Input 1 to select to perform dot product calculation without using theta.	User should be asked to enter inputs for points.	Loading Dot product of two vectors Please select how you would like the dot product calculated: 1: using cosine theta 2: without using cosine theta 2 Not using cosine dot product calculation Please enter X for point A:	yes
10	Entering vector a as 2, 7, 1 and vector b as 3, 5, 0	Should output the result of 41	Please enter X for point A:2 Please enter Y for point A:7 Please enter Z for point A:1 Please enter X for point B:3 Please enter Y for point B:5 Please enter Z for point B:0 a.b = 41	Yes

11	Entering "bob", "yes", "Jack" for vector A and "Bill", "No", "twenty" for vector b	Should display either 0 or NaN	<pre> Not using cosine dot product calculation Please enter X for point A:bob Please enter Y for point A:yes Please enter Z for point A:Jack Please enter X for point B:Bill Please enter Y for point B:No Please enter Z for point B:twenty a.b = 0 </pre>	Yes
12	Entering 0 for all points	Output should be 0	<pre> Not using cosine dot product calculation Please enter X for point A:0 Please enter Y for point A:0 Please enter Z for point A:0 Please enter X for point B:0 Please enter Y for point B:0 Please enter Z for point B:0 a.b = 0 </pre>	Yes
13	Check to see that the user can return to the main menu	User should be redirected to the main menu	<pre> a.b = 0 Returning to main menu. Please select which function you wish to perform from the list below: 1: Vector Length Calculator 2: Normalization of a vector 3: Distance between two points 4: Dot product of two vectors 5: Cross product of two vectors 6: Mean, Mode and Standard Deviation of a list of integers 7: Union and Intersection of two sets Please enter the number for the option you wish to select: </pre>	Yes

Cross product of two vectors testing

Test	Desc.	Expected result	Actual result	Success?
1	Check the user can navigate from the main menu to the cross product calculator.	Navigates to the cross product calculator.	<pre>Please enter the number for the option you wish to select:5 Loading Cross product of two vectors Please enter Vector 0: (</pre>	Yes
2	Enter vector 0 as 2, 7 and 1. The enter vector 1 as 1, -9 and 5.	Result should be (44, 9, -25)	<pre>Loading Cross product of two vectors Please enter Vector 0: (2 ,7 ,1) Please enter Vector 1: (1 ,-9 ,5) This is the X matrix: 7 1 -9 5 This is the Y matrix: 2 1 1 5 This is the Z matrix: 2 7 1 -9 The cross product is: (44,9,-25) Returning to main menu.</pre>	Yes. Side note: Even though cout << endl; is not used here, the vector input is still going to the next line so the appearance of vectors when inputting looks dreadful.

3	Enter 0 for all values	Answer should be 0,0,0	<pre> Loading Cross product of two vectors Please enter Vector 0: (0 ,0 ,0) Please enter Vector 1: (0 ,0 ,0) This is the X matrix: 0 0 0 0 This is the Y matrix: 0 0 0 0 This is the Z matrix: 0 0 0 0 The cross product is: (0,0,0) </pre>	Yes
4	Enter "Bob", "Jack", "Jane", "jill", "Jade", "Roxy" as inputs	Answers should be either 0,0,0 or NaN, NaN, NaN	<pre> Please enter Vector 0: (Bob ,Jack ,Jane) Please enter Vector 1: (jill ,Jade ,Roxy) This is the X matrix: 0 0 0 0 This is the Y matrix: 0 0 0 0 This is the Z matrix: 0 0 0 0 The cross product is: (0,0,0) </pre>	Yes

5	Enter nothing for all inputs	Outputs should be 0,0,0	<pre> Loading Cross product of two vectors Please enter Vector 0: (, ,) Please enter Vector 1: (, ,) This is the X matrix: -9.25596e+61 -9.25596e+61 -9.25596e+61 -9.25596e+61 This is the Y matrix: -9.25596e+61 -9.25596e+61 -9.25596e+61 -9.25596e+61 This is the Z matrix: -9.25596e+61 -9.25596e+61 -9.25596e+61 -9.25596e+61 The cross product is: (0,0,0) </pre>	Partial. The 2x2 matrix's get given a huge negative number.
6	Check that the user is redirected to the main menu once the cross product calculation is complete	User should be moved to the main menu.	<pre> The cross product is: (0,0,0) Returning to main menu. Please select which function you wish to perform from the list below: 1: Vector Length Calculator 2: Normalization of a vector 3: Distance between two points 4: Dot product of two vectors 5: Cross product of two vectors 6: Mean, Mode and Standard Deviation of a list of integers 7: Union and Intersection of two sets Please enter the number for the option you wish to select: </pre>	Yes

Mean, Mode and Standard Deviation testing

Test	Desc.	Expected result	Actual result	Success?
1	Check that the user can navigate to the mean, mode and standard deviation from the main menu	User navigates to the mean mode and standard deviation menu.	Loading Mean, Mode and Standard Deviation of a list of integers. Please select which you wish to pick: 1: Mean calculator 2: Mode calculator 3: Standard Deviation Calculator	Yes
2	Input 0 as a selection	Asked to try again	Loading Mean, Mode and Standard Deviation of a list of integers. Please select which you wish to pick: 1: Mean calculator 2: Mode calculator 3: Standard Deviation Calculator 0 invalid input try again. Please select which you wish to pick: 1: Mean calculator 2: Mode calculator 3: Standard Deviation Calculator	Yes
3	Input 4 as a selection	Asked to try again	Please select which you wish to pick: 1: Mean calculator 2: Mode calculator 3: Standard Deviation Calculator 4 invalid input try again. Please select which you wish to pick: 1: Mean calculator 2: Mode calculator 3: Standard Deviation Calculator	Yes
4	Input "bob" as a selection	Asked to try again	Please select which you wish to pick: 1: Mean calculator 2: Mode calculator 3: Standard Deviation Calculator bob invalid input try again. Please select which you wish to pick: 1: Mean calculator 2: Mode calculator 3: Standard Deviation Calculator	Yes

5	Input 1 to perform mean calculation	Should begin mean calculation	<pre> Please select which you wish to pick: 1: Mean calculator 2: Mode calculator 3: Standard Deviation Calculator 1 Mean Calculator please enter how many numbers will be entered up to 100: </pre>	Yes
6	Input the range for 5, while all items for the mean are 10	$(10*5)/5=10$. 10 should be output as the mean.	<pre> Mean Calculator please enter how many numbers will be entered up to 100:5 Please enter input number 0:10 Please enter input number 1:10 Please enter input number 2:10 Please enter input number 3:10 Please enter input number 4:10 The mean of your values is:10 </pre>	Yes
7	Input 0 for all items	Mean should be 0 or NaN	<pre> Mean Calculator please enter how many numbers will be entered up to 100:0 The mean of your values is:-nan(ind) </pre>	Yes
8	Input bob for range	Mean should be 0 or NaN	<pre> Mean Calculator please enter how many numbers will be entered up to 100:bob The mean of your values is:-nan(ind) </pre>	Yes
9	Input 3 for range and set all other inputs to bob	Mean should be 0 or NaN	<pre> Mean Calculator please enter how many numbers will be entered up to 100:3 Please enter input number 0:bob Please enter input number 1:bob Please enter input number 2:bob The mean of your values is:0 </pre>	Yes
10	Input 2 for selection in order to open mode calculator	Should be asked to enter mode information	<pre> Loading Mean, Mode and Standard Deviation of a list of integers. Please select which you wish to pick: 1: Mean calculator 2: Mode calculator 3: Standard Deviation Calculator 2 Mode Calculator Please enter the range of numbers you will enter: </pre>	Yes

11	Enter a range of 4, with the numbers being 1, 1, 2, 3	Mode should output as 1	<pre> Mode Calculator Please enter the range of numbers you will enter:4 Please enter input number 0:1 Please enter input number 1:1 Please enter input number 2:2 Please enter input number 3:3 Your sorted numbers are: 1 1 2 3 Mode is: 1 </pre>	Yes
12	Enter range as 2, then all inputs as 0	Mode should output as 0	<pre> Please enter the range of numbers you will enter:2 Please enter input number 0:0 Please enter input number 1:0 Your sorted numbers are: 0 0 Mode is: 0 </pre>	Yes
13	Enter range as 0	Mode should be -999999 as it is set to be this value by default	<pre> Mode Calculator Please enter the range of numbers you will enter:0 Your sorted numbers are: Mode is: -999999 Returning to main menu. </pre>	Yes
14	Enter range as 2, while all other inputs are "bob"	Display 0, as the input numbers are being defaulted to 0	<pre> Please enter the range of numbers you will enter:2 Please enter input number 0:bob Please enter input number 1:bob Your sorted numbers are: 0 0 Mode is: 0 </pre>	Yes
15	Select standard deviation in the menu	Ask user about standard deviation inputs	<pre> Loading Mean, Mode and Standard Deviation of a list of integers. Please select which you wish to pick: 1: Mean calculator 2: Mode calculator 3: Standard Deviation Calculator 3 Standard Deviation calculator please enter how many numbers will be entered up to 100: </pre>	Yes

16	Set 7 as the range for standard deviation, entering the numbers 10, 20, 30, 25, 24, 71 and 19.	Should display the mean to be 28.4286, then should square all results, add them together to make 335.1017, and then square root this value to get 18.3057 which will be displayed to the user.	<pre> The mean of your values is:28.4286 Variance for number: 10 is 339.612 Variance for number: 20 is 71.0408 Variance for number: 30 is 2.46939 Variance for number: 25 is 11.7551 Variance for number: 24 is 19.6122 Variance for number: 71 is 1812.33 Variance for number: 19 is 88.898 Standard Deviation is: 48.4326 </pre>	No. Rounding may have been a factor that could have effected either the initial calculation for this test, or the actual output from the code.
17	Set the range as 7 and all values to 0	Output either 0 or NaN.	<pre> The mean of your values is:0 Variance for number: 0 is 0 Variance for number: 0 is 0 Variance for number: 0 is 0 Variance for number: 0 is 0 Variance for number: 0 is 0 Variance for number: 0 is 0 Variance for number: 0 is 0 Standard Deviation is: 0 </pre>	Yes
18	Set the range to 0	Outputs should be 0 or nan	<pre> Standard Deviation calculator please enter how many numbers will be entered up to 100:0 The mean of your values is:-nan(ind) Standard Deviation is: 0 </pre>	Yes
19	Set range to 3 and all outputs to "bob"	Should output 0 or NaN	<pre> please enter how many numbers will be entered up to 100:3 Please enter input number 0:bob Please enter input number 1:bob Please enter input number 2:bob The mean of your values is:0 Variance for number: 0 is 0 Variance for number: 0 is 0 Variance for number: 0 is 0 Standard Deviation is: 0 </pre>	Yes
20	Check that the user gets directed back to the main menu	User should go back to main menu once the standard deviation is calculated.	<pre> Standard Deviation is: 0 Returning to the main menu. Please select which function you wish to perform from the list below: 1: Vector Length Calculator 2: Normalization of a vector 3: Distance between two points 4: Dot product of two vectors 5: Cross product of two vectors 6: Mean, Mode and Standard Deviation of a list of integers 7: Union and Intersection of two sets Please enter the number for the option you wish to select: </pre>	Yes

Union and Intersection of two sets test

Test	Desc.	Expected result	Actual result	Success?
1	Check that the user can navigate from the main menu to Union and Intersection calculator	Union and Intersection is loaded	<pre>Please enter the number for the option you wish to select:7 Loading Union and Intersection of two sets How long is Set A?:</pre>	Yes
2	Setting Set A length to 3, giving the set inputs of 1,2,3. Setting Set B length to 2, giving inputs of 4,5.	Intersect should be displayed as empty, Union will be {1,2,3,4,5}	<pre>How long is Set A?:3 How long is Set B?:2 Please enter set A item number 0:1 Please enter set A item number 1:2 Please enter set A item number 2:3 Please enter set B item number 0:4 Please enter set B item number 1:5 The intersect is as follows: {} The union is as follows: {1,2,3,4,5,}</pre>	Yes
3	Setting Set A to be 3 long, set b to be 2 long, Set A being 1,2,3 while set b is 1,3.	Intersect should be {1,3}, while union should be {1,2,3}	<pre>How long is Set A?:3 How long is Set B?:2 Please enter set A item number 0:1 Please enter set A item number 1:2 Please enter set A item number 2:3 Please enter set B item number 0:1 Please enter set B item number 1:3 The intersect is as follows: {1,3,} The union is as follows: {2,3,3,}</pre>	No. Intersect is correct, however union is displayed completely wrong.

4	Setting both sets range to 0	Not asked for set item inputs, output for intersection and union should be {}	<pre> Loading Union and Intersection of two sets How long is Set A?:0 How long is Set B?:0 The intersect is as follows: {} The union is as follows: {} </pre>	Yes
5	Setting both sets range to "bob".	Should not be asked for set item inputs, union and intersect should be {}	<pre> Loading Union and Intersection of two sets How long is Set A?:bob How long is Set B?:bob The intersect is as follows: {} The union is as follows: {} </pre>	Yes
6	Not entering anything for the range of the sets.	Should display intersect and union as {}	<pre> Loading Union and Intersection of two sets How long is Set A?: How long is Set B?: The intersect is as follows: {} The union is as follows: {} </pre>	Yes
7	Check that the user is redirected back to the main menu after displaying union and intersect.	User should be directed back to the main menu	<pre> The union is as follows: {} Returning to main menu. Please select which function you wish to perform from the list below: 1: Vector Length Calculator 2: Normalization of a vector 3: Distance between two points 4: Dot product of two vectors 5: Cross product of two vectors 6: Mean, Mode and Standard Deviation of a list of integers 7: Union and Intersection of two sets Please enter the number for the option you wish to select: </pre>	Yes

References

- [1] J. Bird, "Vectors," in *Basic Engineering Mathematics*, Oxford, Taylor & Francis LTD - M.U.A, 2010, p. 266.
- [2] A. Croft and R. Davison, "The Scaler Product," in *Foundation Maths 5e*, Harlow, Pearsons Education M.U.A, 2010, p. 323.
- [3] MathsIsFun.com, "Distance Between 2 Points," MathsIsFun.com, [Online]. Available: <https://www.mathsisfun.com/algebra/distance-2-points.html>. [Accessed 24 April 2020].
- [4] The Organic Chemistry Tutor, "Cross Product of Two Vectors Explained!," 29 April 2017. [Online]. Available: <https://www.youtube.com/watch?v=pWbOisq1MJU>. [Accessed 28 April 2020].
- [5] MathsIsFun, "Standard Deviation and Variance," MathsIsFun.com, [Online]. Available: <https://www.mathsisfun.com/data/standard-deviation.html>. [Accessed 28 April 2020].

Application of Maths for Software Development

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

72/100

PAGE 1



Comment 1

an example is...

PAGE 2

PAGE 3

PAGE 4



Comment 2

next time please ask/email and we can go through it again with examples.

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

GRADING FORM: SMU GENERIC_GJ

MAY DAVIES**72****RESEARCH-INFORMED LITERATURE**

Some very good work here. Glad to see you used the online texts available from library - Bird.

4**KNOWLEDGE AND UNDERSTANDING**

Some very good work here. Please do ask if you are unsure.

14**ANALYSIS**

Good analysis of problem. A detailed test plan generated from the analysis of the problem.

15**PRACTICAL APPLICATION AND DEP.**

Some good work here but some problems in implementation. Overall a very good attempt here. A

35

lot of effort input in testing.

SKILLS FOR PROFESSIONAL PRAC.



References appropriate. Report structured.

4