

Class No 2

ENG2002 COMPUTER PROGRAMMING  
APPLICATION DEVELOPMENT ASSIGNMENT REPORT

Task No 2 of Assignment 3

LIU Yuzhong  
ZHAO Yifei

## Abstract

We aimed not only to finish the assignment requirements, i.e., doing two computations of vectors inputted by users but also to set a humanised login menu function for users. At last, we successfully built up the programme using the techniques taught in the lessons. The programme is humanised enough, which allows the user to choose the function that he or she wants to use or quit the programme on the first menu after successfully logging in.

### 1. Introduction

The main objective of this project is to provide a system in which the user can input two 3D vectors, compute their dot products and the arctan function using the required formula.

The assignment requires us to design a class named `vecThreeD`, representing a 3D vector. The member functions of this class allow us to compute the dot product of two given vectors and calculate the arctan function of a 3D vector. The class and the implementation of the member functions of the class `vecThreeD` is in a static library, which is linked to the console application. Additionally, we are required to implement a mechanism that will ask the user to key in the username. It will be checked with a file storing username-password pairs, and the program will ask the user to key in the password to create a new account if the provided username cannot be found in the file. Otherwise, the user should key in the corresponding password with at most three attempts before the user is knocked out. After the user successfully logs in, he (or she) can select from the main menu that will pop up, to indicate what operation he (or she) likes to conduct.

To achieve our final objective, we develop and test every function, respectively (e.g., the login () function, the arctan () function) before putting them together. The main operations on the 3D vectors, such as the input of vectors, computation of the dot products of two 3D vectors and arctan function are implemented in the class, which is in the static library. The console application is responsible for providing the login system and the main menu (), which acts as a guideline for the user to key in.

### 2. Methodology

#### 1) Division of Work

Both of us have participated actively in the project and exchanged our ideas when developing it. ZHAO Yifei developed the basic structure of the console application, after which LIU Yuzhong designed the method of passing the object of class “`vecThreeD`” by reference to the menu () function. We cooperated to develop the login () function and the header file of the static library. The implementation of `calVec ()` function in the class

“vecThreeD”, which is the main objective of the class, is also the result of our cooperation. In terms of the remaining parts of the implementation of the member functions of “vecThreeD”, LIU Yuzhong is responsible for implementing the exp () and arctan () functions. And the arctan () function was later modified by ZHAO Yifei before the submission.

To sum up, both of us have been closely cooperating with each other throughout the whole process, putting forward methods of optimisation and modifying each other’s codes.

## 2) Schedule and Steps

In the first week when the assignment was uploaded, we write the codes of the calculation of the dot product. Then, we finished the arctan calculation part in the second week. In the last two weeks, we improved the dot product calculation parts and created the static library. At the same time, we developed the login function and implemented it into the source file of the main application.

Before creating the static library, firstly, we tried to implement the function in the source file to read the input value from the user, store the value and do the dot product calculation without using classes. When the project was built successfully, we tried to create the class and put the read, display and calculation functions inside. Up to that time, we were using global variables in the source file to store the coefficients of the vectors. Then we successfully finished the calculation of arctan values with the help of for loop.

Later, when we began to create the static library, we felt that it’s no good to set the variables that store the coefficient as global variables in the source file of the application, so we tried to improve our previous work. We put the variables inside the private part of the class and implemented 6 “getVec” functions to get the value of the variables. Also, in the calculation part, we managed to pass the parameters by reference to the calculation functions to make use of what we had learned to perfect the programme.

Lastly, we created the static library with the class finished in the source file. At the same time, we began to use the knowledge about file input and output taught in the last two lectures and created the login function. We implemented the static library into our main application and added the login function into the source file. Then we modified the codes in main () to make sure the programme goes smoothly as designed. Finally, we tested the output of the programme, made sure there is no error and finished our report.

### 3) Details of the Application

- Specifications of Classes defined and Public/Private Member Functions/Variables Inside

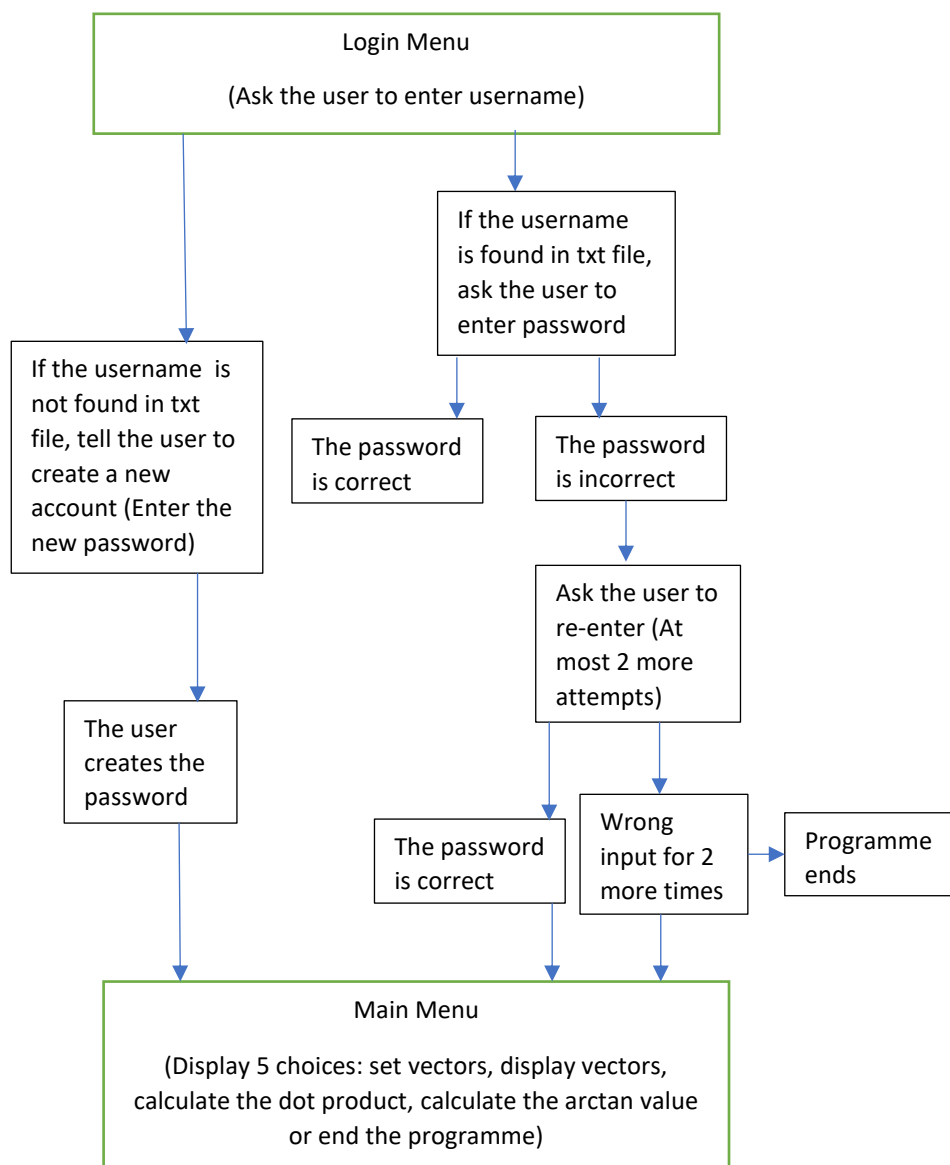
The class `vecThreeD` we defined is used to represent 3D vectors and do the required operation on vectors. It consists of six private member variables: `o`, `p`, `q`, `r`, `s`, `t`, which denote the elements of the two 3D vectors respectively (`o`, `p` and `q` denote the three elements of the first vector and `r`, `s` and `t` for the second vector). In terms of the public member functions, it has six similar functions (`getVec1()` to `getVec6()`) to return the elements of the two vectors respectively, and another six functions (`setVec1(float)` to `setVec6(float)`) to set the value of the six elements of the two vectors respectively. Besides, there is a `calVec()` public member function to compute the dot product of the two 3D vectors, an `exp (double, int)` to calculate exponential function and an `arctan (double, double, int)` to compute the arctan value of the vector. Besides, the class `vecThreeD` contains a public member function `displayVec()` to display the two 3D vectors on the output screen. It is worth mentioning that the class includes a constructor as well, which initialise the six private member variables to 0.

The main objective of this program is to conduct the operation of computing the dot product of two 3D vectors. Thus, it is necessary to provide several variables to denote the elements of the vectors, which is 6, in our case. That is why we set six private member variables. Since the variables are private, which can be accessed by class public member functions, so we proceed to create a set of public member functions, including `setVec1()` to `setVec6()` and `getVec1()` to `getVec6()` to set the values entered by the user to the private variables and to get the private variables as well. In terms of the `calVec()` member function, it is a must if we are to compute the dot products of two 3D vectors. Another objective is to compute the arctan function of the 3D vector; that is the reason we implement a function called `arctan ()`. However, unlike python, C++ does not have a built-in function to calculate the exponential function, which is required when computing the arctan function. To illustrate, the formula is like this:  $\arctan(z) = z - \frac{z^3}{3} + \frac{z^5}{5} - \dots + (-1)^{(\text{term}-1)} \frac{z^{(2*\text{term}-1)}}{(2*\text{term}-1)}$ . Thus, we decided to create a function named `exp ()` to compute the exponential function. Furthermore, with the aim of testing the input results and providing the user with a clearer overview of the entered 3D vectors, we create a member function called `displayVec()`, which will display the two 3D vectors on the output screen. Last but not least, we implement a constructor for the class `vecThreeD` to initialise the six private variables, preventing run-time errors in the case where the user calls the `displayVec()` before entering data into the 3D vectors.

- Flow of Execution
  - i. Login Part

The login menu first asks the user to enter the username. After the user enters, the programme checks if the username can be found in the login\_info.txt. If the username is new, the programme asks the user to create a password and let the user go to the main menu. If the username exists before, the user will be asked to enter the password. If the password is wrong three times, the programme ends. If the password is correct within two attempts, the programme lets the user enter the main menu.

Flowchart:



## ii. Calculator Part

The programme first asks the user to choose the function from the five choices. If the user decides to set the vectors, he (or she) then has to determine which vector to set (1 or 2). The user can set one vector at a time. After the user sets the vector, it will go back to the main menu.

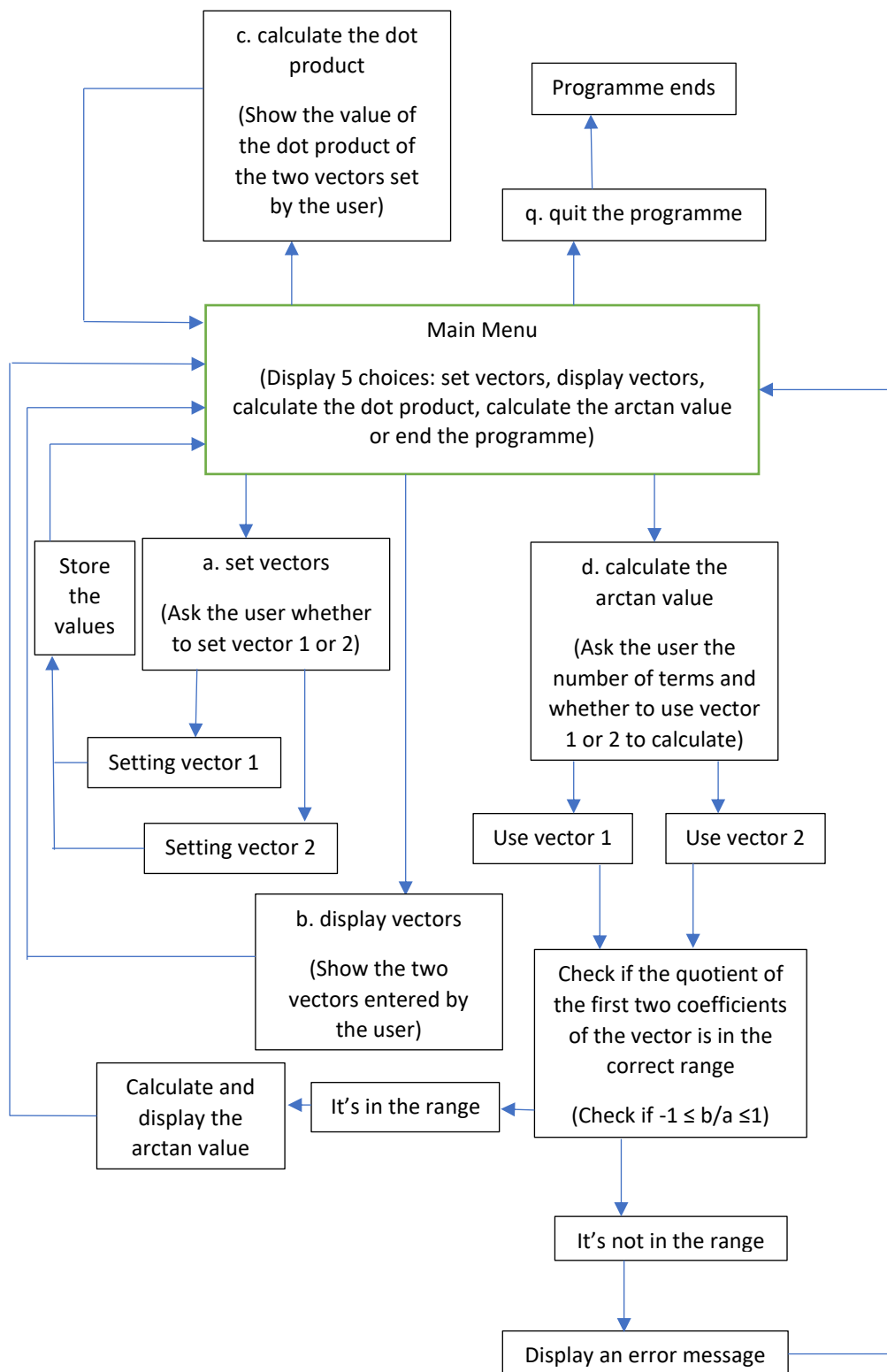
If the user selects to display the vectors, the two vectors will be displayed (in the format of  $a_i + b_j + c_k$ ). The initial values of the coefficients are all zero.

If the user chooses to calculate the dot product of the two vectors, the dot product of vector 1 and 2 will be displayed on the screen.

If the user chooses to calculate the arctan value of the two vectors, he (or she) then has to enter the number of terms and select which vector to calculate. The programme will first check if the quotient ( $z$ ) of the first two coefficients ( $b/a$ ) is between -1 and 1. If  $z$  is out of the range, the programme will display an error message and return to the main menu. If  $z$  is in the correct range, the value of the arctan value will be displayed.

If the user chooses to quit, the programme ends.

Flowchart:



#### 4) Problems Encountered

During the developing process, we've encountered many problems, but we managed to fix them all at last.

Initially, we decided to use six global variables to denote the six elements of the two 3D vectors. However, we discovered it is a better idea to set the six global variables to the private variables of the class `vecThreeD`. Not only can this approach prevent the six variables being assessed and modified randomly, but also it provides a better opportunity for potential changes of the program into a more complex system.

Another major setback we've encountered is about the declaration of the object of class `vecThreeD`. After building our first version of the program, we tried to run it, when we found that no matter what numbers we keyed in, both vectors remained  $0i+0j+0k$ , which is the default value. To find why the run-time error occurred, we used the run-time debugger to debug the program. It led us to the root of the problem: we declared the object of the class `vecThreeD` in the `menu ()` function, which means that a new object was declared every time the loop executed. We solved this problem by simply putting the declaration of the object into the `main ()` function.

We've also met a problem concerning the login function in the application. When validating the results, i.e., when entering the username we had set before, we found that the programme couldn't recognise the username every time. Later, suddenly, when we deliberately entered a wrong username (i.e., the correct user name was "dddd", and we entered "ddd"), we logged in successfully with the corresponding password. We got confused about why the programme could recognise part of the username in the txt file from time to time. We searched the Net, reviewed lecture notes and videos, and sought help from classmates and teachers. Finally, we understood that the while loop would read one character from `fin` so that the first username will miss the first character, that's why it suddenly recognised "ddd" from the txt file. So we added `tellg()` and `seekg()` function to move the cursor backwards for one position to let the programme read the complete user name. Also, when reading the username, we had to replace `cin.get()` or `cin.getline()` with `fin>>` since the space between the username and password will prevent the password from being read to the username. With the help from the teacher and our better understanding, we finalised the login function part.

#### 5) Validation of the Application

For login function part, at first, the `login_info.txt` file does not exist. So we run the programme first and create some username-password pairs. We then opened the txt files and checked if the username-password pairs were stored correctly according to our input. Next, we opened the programme again and entered the username and checked if we could go smoothly to be allowed to enter the password. Confirming everything was right, we entered the password and saw if we could go to the main menu. The detailed results will be shown in the 3<sup>rd</sup> part of the report.

In the main menu, we first chose b to display the current value of the two vectors to



confirm that the coefficients were all zero. Then the main menu will appear again, and we chose a to set the coefficients of the two vectors. After that, we displayed the vectors again and made sure that the values were stored correctly. And we calculated the dot product ourselves by using a calculator. After all of these were done, we chose c to calculate the dot product of the two vectors. We compared the returned value to the value previously gained and saw if I was correct.

For the arctan value part, we set another pair of vectors. Vector 1's value (b/a) will be in the correct range (from -1 to 1) while vector 2's value will be out of the range. We chose d twice to calculate the arctan values of each vector and noted the results. For vector 1, it should return the value of the arctan value, and we compared it to the value gained by using a calculator. For vector 2, it should return the error message set before. The detailed results of the two calculations will also be shown in the 3<sup>rd</sup> part of the report.

### 3. Results

#### Login function

#### Creating Accounts

The image shows two side-by-side screenshots of the '3D-Vector Calculator' application. Both windows show the same sequence of events: a 'Welcome' message, a prompt to enter a username (which is not found), a prompt to enter a password (which is 'Handsome'), a 'New account created' message, and a 'Welcome' message for the new user. The first window shows the user 'FrankIs' and the second shows 'JackIs'. Both windows then display a 'Main Menu' with options: [a] Input your vectors, [b] Display your vectors, [c] Calculate the dot product, [d] Calculate the arctan value, and [q] Quit. The prompt 'Enter your choice:' is visible at the bottom of each window.

The image shows a Notepad window titled 'login\_info.txt'. The text inside the window is as follows:  
 FrankIs Handsome  
 JackIs Handsome

Re-open the programme (Correctly entering the password; Wrong input for three times; Wrong input for the first time but input correctly later)

The image shows two side-by-side screenshots. The left window shows the '3D-Vector Calculator' application with the user 'FrankIs' successfully logging in after entering the password 'Handsome'. The right window shows the 'Microsoft Visual Studio Debug Console' output for the same program. It shows the user 'FrankIs' entering the password 'Smart', which is incorrect. The program outputs 'Wrong password! 2 attempt(s) left.' and prompts the user to re-enter the password. The user enters 'Jeng', which is also incorrect, resulting in 'Wrong password! 1 attempt(s) left.' and another prompt. Finally, the user enters 'Awesome', which is also incorrect, resulting in the message: 'You've entered incorrect passwords for too many times, the programme ends!'. The console also shows the file path and the process exiting with code 0.

```
C:\Users\dylan\OneDrive - The Hong Kong Polytec...
Welcome to 3D-Vector Calculator!
Please enter the username (spaces are not allowed): JackIs
Password: NothHandsome
Wrong password! 2 attempt(s) left.
Please re-enter your password: VeryHandsome
Wrong password! 1 attempt(s) left.
Please re-enter your password: Handsome
Welcome JackIs!
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice:
```

Main Function

Check the initial value of the vectors

```
C:\Users\dylan\OneDrive - The Hong Kong Polytech...
Welcome to 3D-Vector Calculator!
Please enter the username (spaces are not allowed): FrankIs
Password: Handsome
Welcome FrankIs!
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice: b
Vector 1 is: 0i + 0j + 0k.
Vector 2 is: 0i + 0j + 0k.
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice:
```

Set the two vectors to  $2i+1j+4k$  and  $1i+2j+4k$  respectively

```
C:\Users\dylan\OneDrive...
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice: a
If you want to input vector 1, press 1.
If you want to input vector 2, press 2.
Enter your choice: 1
Key in a: 2
Key in b: 1
Key in c: 4
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice: a
If you want to input vector 1, press 1.
If you want to input vector 2, press 2.
Enter your choice: 2
Key in a: 1
Key in b: 2
Key in c: 4
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice: b
Vector 1 is: 2i + 1j + 4k.
Vector 2 is: 1i + 2j + 4k.
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice:
```

Calculate the dot products of the two vectors ( $2*1+1*2+4*4=20$ , so the result should be 20)

```

C:\Users\dylan\OneDrive - The ...
Enter your choice: b
Vector 1 is: 2i + 1j + 4k.
Vector 2 is: 1i + 2j + 4k.
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice: c
The dot product of the two vectors is: 20.
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice:

```

Calculate the arctan value for vector 1 [Set the number of terms at 5, for vector 1,  $b/a = \frac{1}{2}$ ,  $\arctan(b/a)=0.463648$  (by calculator), and the value should be  $0.5 - (0.5^3)/3 + (0.5^5)/5 - (0.5^7)/7 + (0.5^9)/9 = 0.463684$  (by the formula given in the instructions), so the programme should return 0.463684]

```

C:\Users\dylan\OneDrive - The Hong Kong Polytechnic Univ...
Enter your choice: 2
Key in a: 1
Key in b: 2
Key in c: 4
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice: b
Vector 1 is: 2i + 1j + 4k.
Vector 2 is: 1i + 2j + 4k.
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice: d
Enter the number of terms: 5
Enter the number of vector that you want to calculate (1 or 2):1
The value is: 0.463684
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice:

```

Calculate the arctan value for vector 2 (Since  $a = 1$  and  $b = 2$  here,  $b/a = 2$  is out of the range, the programme should display an error message.)

```

C:\Users\dylan\OneDrive - The Hong Kong Polytechnic Univ...
Enter your choice: b
Vector 1 is: 2i + 1j + 4k.
Vector 2 is: 1i + 2j + 4k.
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice: d
Enter the number of terms: 5
Enter the number of vector that you want to calculate (1 or 2):2
Your input is not in the required range!
=====Main Menu=====
[a] Input your vectors
[b] Display your vectors
[c] Calculate the dot product
[d] Calculate the arctan value
[q] Quit
Enter your choice:

```

#### 4. Conclusion and further development

##### 1) Experience Gained

After having done the application development assignment, we have a better understanding of the knowledge taught during the lectures and know the principles of C++ more profound.

We developed skills in setting a class and its member functions and variables, building a static library and using file i/o stream to set up login functions. Instead of using global variables and passing the parameters by value, we developed our application, and we applied the knowledge of pass by reference in the calculations, which also helps us strengthen our understanding of pointers and address.

When developing the application, we've learned about the importance of collaboration. Neither of us could handle the task on one's own, intense cooperation between us is tremendously significant. When encountering bugs and run-time errors, LIU Yuzhong or ZHAO Yifei often couldn't find the mistake in the codes alone. But when he called the other groupmate to help solve the problem, the two students could see the errors and debug the programme much more quickly. This experience teaches us a lesson: people with the same level of knowledge may have completely different mindsets, which demonstrates the necessity and significance of collaboration.

##### 2) Possible Extension and Improvements

If we could have more time, maybe we would try to develop the logic of the first main menu, especially for the setting vectors part. We might use another "if" condition to finish the following:

For the set vector function, we could let the user set two vectors at one time. To be more specific, after the user setting vector 1, the programme can ask if he (or she) wants to set vector 2 as well. If the user chooses yes, then the user can set vector 2 and vice versa. It needs more lines of codes using maybe if conditions.

Furthermore, if we could have more time, we might also further develop the login function. Specifically, after the user creates a new login account, we could add some loops in the codes (maybe using "if" condition) so that the application could go back to login function to let the user enter username and passwords again after creating new accounts.

Last but not least, if we could have more time, we might develop a function to let the user specify the number of decimal places for the calculation and the returned value will contain the exact number of the decimal places set by the user. We speculate that this function can be achieved by function overloading. However, this part has not been fully covered in the lectures so we might have to spend more time searching on the Net for information and knowledge concerned if we want to do so.

The End.