

Developmental Testing

User Interface

Quotes Menu

1. Add Quote
2. Change quote information
3. Delete quote
4. View quote
5. Return to Main Menu

Enter choice: 5

I found the error that when I attempted to return from the quotes menu to the main menu it did not take me.

```
int quotestatus=0;
while(quotestatus != 4)
{
    quotestatus = QuoteMenu();
} //endwhile
getch();
return 0;
```

This was because I accidentally entered 4 instead of 5 within the while loop so did not break the loop.

Quotes Menu

1. Add Quote
2. Change quote information
3. Delete quote
4. View quote
5. Return to Main Menu

Enter choice:

```
[
int quotestatus=0;
while(quotestatus != 5)
{
    quotestatus = QuoteMenu();
} //endwhile
getch();
return 0;
]
```

Installation

- I was given the error that one of my variables for the staff file was undefined.
- I soon found that the error was a mistype, I declared at the top it was one d in the variable however in the program I used two.
- To resolve this, I removed the extra d from the address variables.

```

    fin.getline(lnamestaff[count], sizeof(lnamestaff[count]));
    fin.getline(oneadstaff[count], sizeof(oneadstaff[count]));
    fin.getline(twoadstaff[count], sizeof(twoadstaff[count]));
    fin.getline(threaddstaff[count], sizeof(threaddstaff[count]));
    fin.getline(pcodestaff[count], sizeof(pcodestaff[count]));
    fin.getline(telnostaff[count], sizeof(telnostaff[count]));
    fin.getline(ninum[count], sizeof(ninum[count]));
    fin.getline(username[count], sizeof(username[count]));
    fin.getline(password[count], sizeof(password[count]));
    fin.getline(loa[count], sizeof(loa[count]));
}
fin.close();
return 0;
}

int Customers ()
470: 32 Modified Insert \Code/
C++ Warning] Prototype.cpp(310): W8070 Function should return a value
C++ Error] Prototype.cpp(470): E2451 Undefined symbol 'oneadstaff'
C++ Error] Prototype.cpp(470): E2109 Not an allowed type
C++ Error] Prototype.cpp(470): E2285 Could not find a match for 'istream::getline(undefined unsigned int)'
ild/

```

```

char oneadstaff[10][30];
char twoadstaff[10][15];
char threaddstaff[10][15];

```

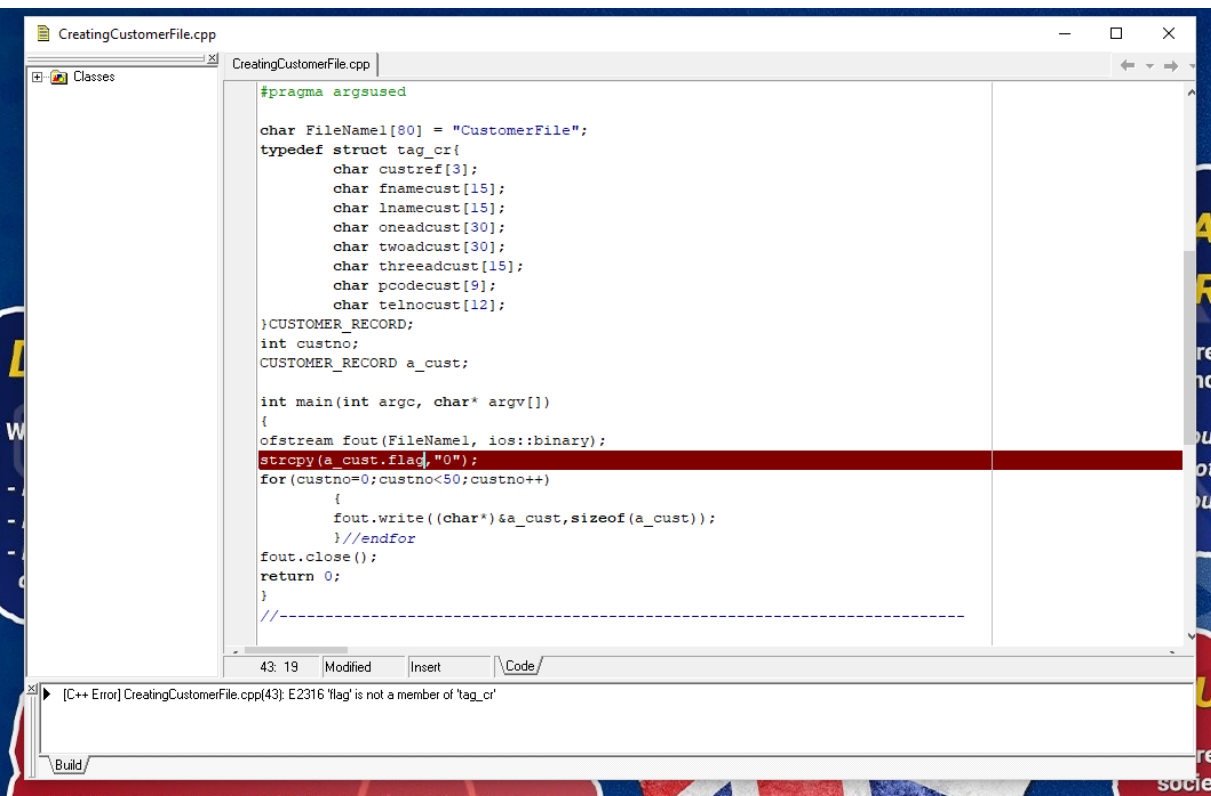
Error

```

fin.getline(oneadstaff[count], sizeof(oneadstaff[count]));
fin.getline(twoadstaff[count], sizeof(twoadstaff[count]));
fin.getline(threaddstaff[count], sizeof(threaddstaff[count]));

```

Fix



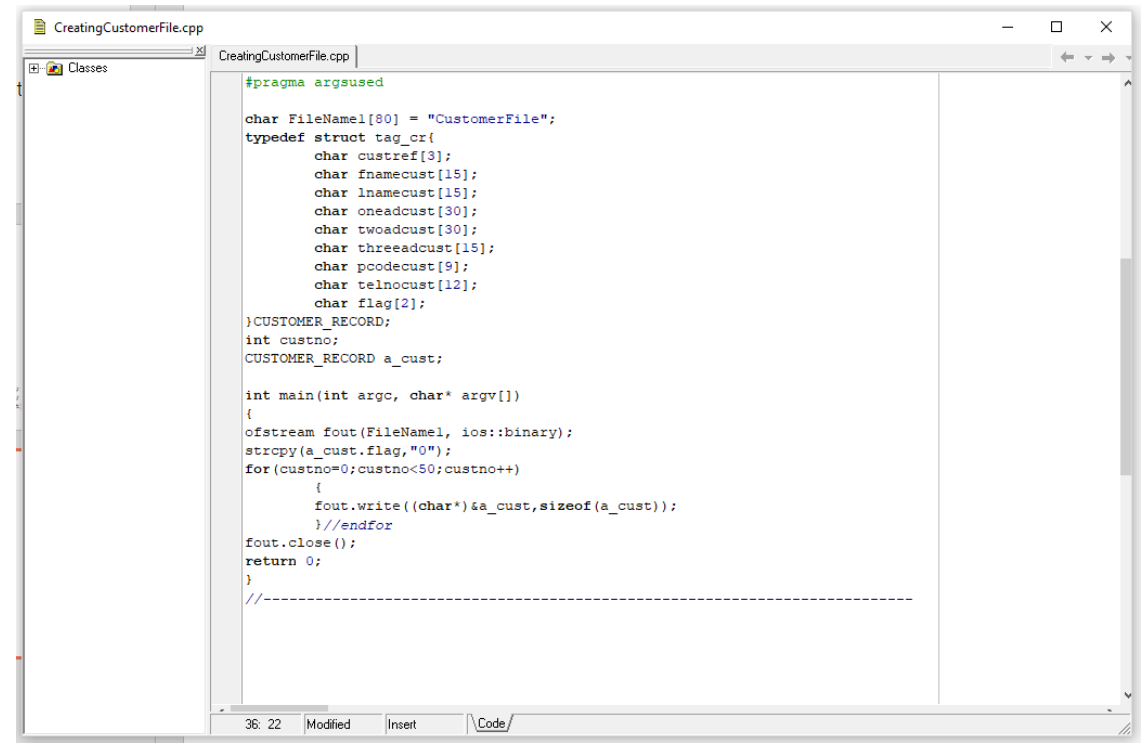
```
#pragma argsused

char FileName1[80] = "CustomerFile";
typedef struct tag_cr{
    char custref[3];
    char fnamecust[15];
    char lnamecust[15];
    char oneadcust[30];
    char twoadcust[30];
    char threeadcust[15];
    char poodecust[9];
    char telnocust[12];
}CUSTOMER_RECORD;
int custno;
CUSTOMER_RECORD a_cust;

int main(int argc, char* argv[])
{
    ofstream fout(FileName1, ios::binary);
    strcpy(a_cust.flag, "0");
    for(custno=0; custno<50; custno++)
    {
        fout.write((char*)&a_cust, sizeof(a_cust));
    } //endfor
    fout.close();
    return 0;
}
//-----

43: 19 Modified Insert Code
```

[C++ Error] CreatingCustomerFile.cpp(43): E2316 'flag' is not a member of 'tag_cr'



```
#pragma argsused

char FileName1[80] = "CustomerFile";
typedef struct tag_cr{
    char custref[3];
    char fnamecust[15];
    char lnamecust[15];
    char oneadcust[30];
    char twoadcust[30];
    char threeadcust[15];
    char poodecust[9];
    char telnocust[12];
    char flag[2];
}CUSTOMER_RECORD;
int custno;
CUSTOMER_RECORD a_cust;

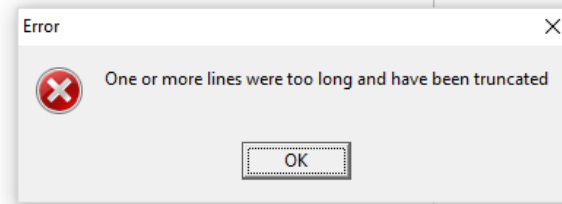
int main(int argc, char* argv[])
{
    ofstream fout(FileName1, ios::binary);
    strcpy(a_cust.flag, "0");
    for(custno=0; custno<50; custno++)
    {
        fout.write((char*)&a_cust, sizeof(a_cust));
    } //endfor
    fout.close();
    return 0;
}
//-----

36: 22 Modified Insert Code
```

I was given the error that the flag was not defined. I quickly found that the error was I had not included the flag as a variable. Once I did this as shown in the photo on the right the program ran correctly.

Customers

```
Add Customer*****
Enter customer reference: 2
```



I found that when I ran the program it wasn't allowing me to add the customer and once the program was closed an error message appeared.

This was because when a user was adding the information it was storing each variable as 80 characters instead of the size I declared at the top. This means that the file was far too large.

Once these were change the program ran correctly and added the customer information to the random access file

```
Prototype.cpp
{
    cout<<"\nAdd Customer";
    cout<<"*****";
    int compare;
    cout<<"\nEnter customer reference: ";
    cin>>custref;
    ifstream fin(FileNamel,ios::binary);
    fin.seekg(custref*sizeof(a_cust));
    fin.get((char*)&a_cust,sizeof(a_cust));
    fin.close();
    compare = strcmpi(a_cust.flag, "0");
    if(compare == 0)
    {
        cout<<"IF";
        ofstream fout(FileNamel,ios::in);
        fout.seekp(custref*sizeof(a_cust));
        cin.get();
        cout<<"\nEnter first name: ";
        cin.getline(a_cust.fnamecust,80);
        cout<<"\nEnter last name: ";
        cin.getline(a_cust.lnamecust,80);
        cout<<"\n";
        cout<<"\nEnter address line 1: ";
        cin.getline(a_cust.oneadcust,80);
        cout<<"\nEnter address line 2: ";
        cin.getline(a_cust.twoadcust,80);
        cout<<"\nEnter address line 3: ";
        cin.getline(a_cust.threadcust,80);
        cout<<"\nEnter postcode: ";
        cin.getline(a_cust.pcodecust,80);
        cout<<"\n";
        cout<<"\nEnter telephone number: ";
        cin.getline(a_cust.telnocust,80);
        strcpy(a_cust.flag,"1");
    }
}
```

```
{
    ofstream fout(fileName2,ios::in);
    fout.seekp(custref*sizeof(a_cust));
    cin.get();
    cout<<"\nEnter first name: ";
    cin.getline(a_cust.fnamecust,15);
    cout<<"\nEnter last name: ";
    cin.getline(a_cust.lnamecust,15);
    cout<<"\n";
    cout<<"\nEnter address line 1: ";
    cin.getline(a_cust.oneadcust,15);
    cout<<"\nEnter address line 2: ";
    cin.getline(a_cust.twoadcust,15);
    cout<<"\nEnter address line 3: ";
    cin.getline(a_cust.threadcust,15);
    cout<<"\nEnter postcode: ";
    cin.getline(a_cust.pcodecust,9);
    cout<<"\n";
    cout<<"\nEnter telephone number: ";
    cin.getline(a_cust.telnocust,12);
    strcpy(a_cust.flag,"1");
    fout.write((char*)&a_cust,sizeof(a_cust));
    fout.close();
}
getch();
return 0;
}
```

Prototype.cpp				
C	17	Fir	Bei	s


```
cout<<"\nAdd Customer";
cout<<"\n*****";
cout<<"\nEnter new customer reference: ";
cin>>custref;
```

```
ifstream fin(FileName2,ios::binary);
fin.seekg(custref*sizeof(a_cust));
fin.get((char*)&a_cust,sizeof(a_cust));
fin.close();
compare = strcmpi(a_cust.flag, "1");
if(compare == 0)
{
```

```
compare = strcmpi(a_cust.flag, "0");
if(compare == 0)
```

I had an error that when I entered a new customer reference to add a customer it wouldn't allow it. Even when the customer file was empty it was still saying the customer reference was in use and for customer references where data is already stored it was allowing more information to be entered.

When I looked back at the code it became clear I had used the wrong value for the comparison of the flag. I was comparing it to 1 which is for when data is stored however I needed to check that the position of the file was empty so I should have been comparing to 0.

```
H:\Y12\Computer Science\Coursework\4 - Prototy
View customer by name
*****
Enter customers last name: Smith
Full name:
Address line 1:
Address line 2:
Address line 3:
Postcode:
Telephone number:
Full name:
Address line 1:
Address line 2:
Address line 3:
Postcode:
Telephone number: _
```

Once I run this and attempted to view a customer stored in the file it outputted as all records in the file to be empty. This was because of a `cin.get()` that I added making it seem like when the name was entered nothing was there.

```
cout<<"\n*****";
int compare;
char name[25];
cout<<"\nEnter customers last name: ";
cin.getline(name,25);
cin.get();
```

```
int compare;
char name[25];
cout<<"\nEnter customers last name: ";
cin.get();
cin.getline(name,25);
```

I then moved the `cin.get()` and the program ran correctly only bringing up the correct customers details.

```
Change Customer Home Address
*****
Enter last name of the customer: Smith

Enter address line 1:
Enter address line 2:
```

```
cout<<"\nEnter last name of the customer: ";
cin.getline(name,15);
cin.get();

for(custref=0;custref<10;custref++)
{
    cout<<"\n Name in loop: "<<name;
    ifstream fin(fileName2,ios::binary);
    fin.seekg(custref*sizeof(a_cust));
    fin.get((char*)&a_cust,sizeof(a_cust));
    fin.close();
}
```

```
Change Customer Home Address
*****
Enter last name of the customer: Smith

Name in loop:
Enter address line 1:
Enter address line 2: _
```

When I attempted to change the home address of the customer I had the error that it was not saving an changes to the file and was also skipping over the first line of the address so no changes could be made.

Firstly, I used a dummy print inside the loop to see if the name entered by the user was actually being saved since there was no change in the file so no link.

Once I did this, it showed that it was not saving the name so when reaching the loop to add it was blank.

After looking through my code I saw that I had written a `cin.get` underneath where the name is entered and thought that this may be causing the issue so commented it out.

This then saved the name and was used in the loop however it was still skipping the first line of the address. To solve this issue, I moved the `cin.get` above where the customer name is entered and the program ran correctly.

DevTestProgram.cpp	CustomerFileSD						
	Fion	McReynolds	79 Stoney Bank Drive	Kiveton Park	Sheffield	S26 6SB	07529729498 1

```
Delete a customer
*****
Enter customer reference: 2

Full name: Fion McReynolds
Mobile number: 07529729498
Is this the correct customer to be deleted (0=Yes, 1=No): 0
```

DevTestProgram.cpp	CustomerFileSD						
	Fion	McReynolds	79 Stoney Bank Drive	Kiveton Park	Sheffield	S26 6SB	07529729498 1

When attempting to delete a customer I found the value of the flag at the end of the record wasn't changing once I confirmed it to be deleted.

Looking back at the code I check that the confirmation was returning the correct value to the variable 'result' which it was, which meant that it was getting to the code for deleting.

After looking at the section of code which deletes it from the file I found that the value of the flag was being saved as 1 rather than 0 meaning the code could not actually be overwritten. Once this was changed, the customer was deleted successfully.

Continues...

```

if(compare==0)
{
    cout<<"\nFull name: "<<a_cust.fnamecust<<" "<<a_cust.lnamecust;    //
    cout<<"\nMobile number: "<<a_cust.telnocust;
    cin.get();
    cout<<"\nIs this the correct customer to be deleted (0=Yes, 1=No): ";
    cin>>result;
    if(result == 0)
    {
        ofstream fout(fileName2,ios::in);
        fout.seekp(custref*sizeof(a_cust));
        strcpy(a_cust.flag,"1");          //Location is now
        fout.write((char*)&a_cust,sizeof(a_cust));
        fout.close();
    }// end if
} //end if

```

Code before change

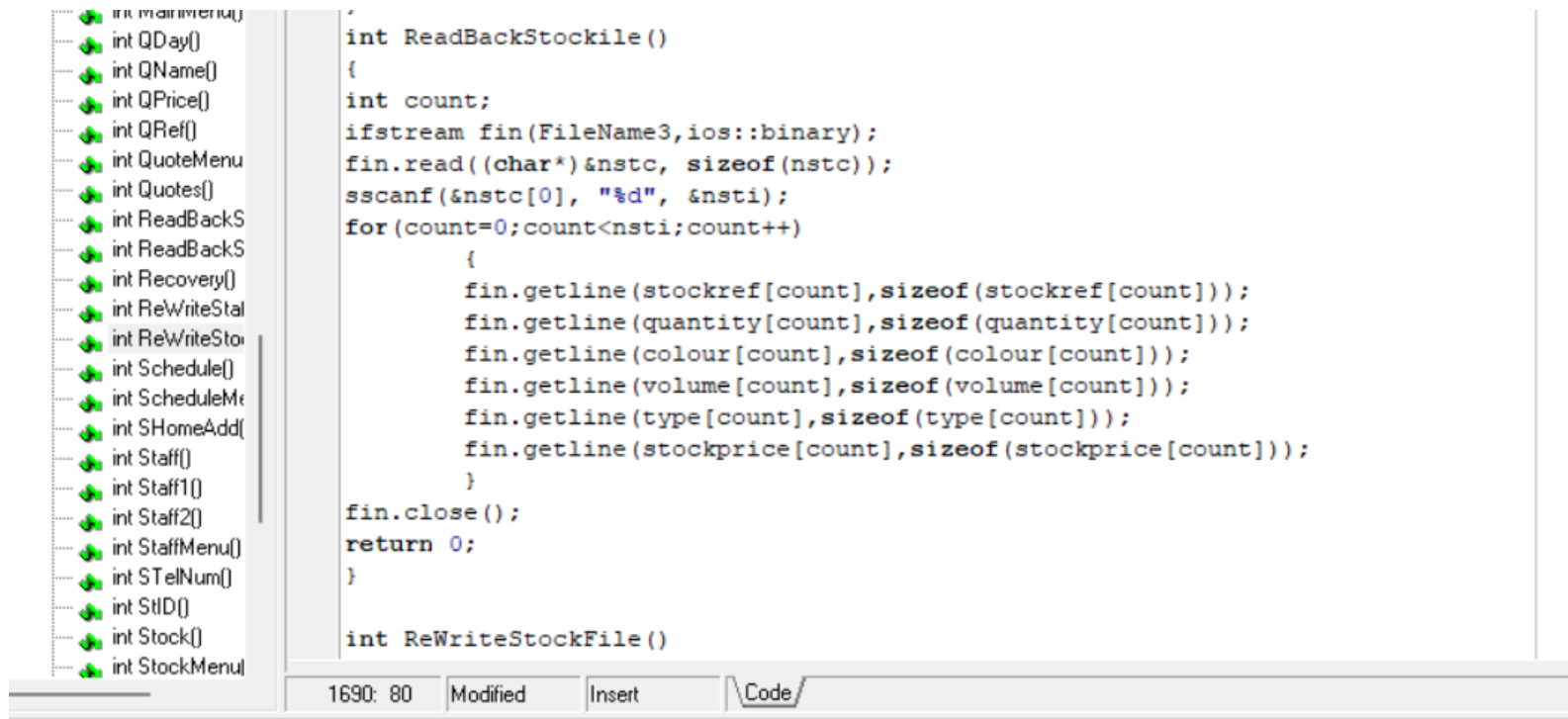
```

if(compare==0)
{
    cout<<"\nFull name: "<<a_cust.fnamecust<<" "<<a_cust.lnamecust;
    cout<<"\nMobile number: "<<a_cust.telnocust;
    cin.get();
    cout<<"\nIs this the correct customer to be deleted (0=Yes, 1=No): ";
    cin>>result;
    if(result == 0)
    {
        ofstream fout(fileName2,ios::in);
        fout.seekp(custref*sizeof(a_cust));
        strcpy(a_cust.flag,"0");          //Location is
        fout.write((char*)&a_cust,sizeof(a_cust));
        fout.close();
    }// end if
} //end if

```

Code after change

Stock



```
int ReadBackStockFile()
{
    int count;
    ifstream fin(FileName3,ios::binary);
    fin.read((char*)&nstc, sizeof(nstc));
    sscanf(&nstc[0], "%d", &nsti);
    for (count=0;count<nsti;count++)
    {
        fin.getline(stockref[count],sizeof(stockref[count]));
        fin.getline(quantity[count],sizeof(quantity[count]));
        fin.getline(colour[count],sizeof(colour[count]));
        fin.getline(volume[count],sizeof(volume[count]));
        fin.getline(type[count],sizeof(type[count]));
        fin.getline(stockprice[count],sizeof(stockprice[count]));
    }
    fin.close();
    return 0;
}

int ReWriteStockFile()
```

[Linker Error] Unresolved external 'ReadBackStockFile()' referenced from C:\USERS\FIONM\ONEDRIVE\DOCUMENTS\PROTOTYPE\PROTOTYPE\UNIT1.OBJ

When I attempted to run the program, an error was produced telling me the 'ReadBack' function was not working.

I soon found out that I has mistyped the name of the function in the program, once I changed this the program ran correctly.

```
int ReadBackStockFile()
{
```

When I ran my sort of the stock quantities from low to high the quantities were not actually been sorted.

This was because I read back my stock file however after did nothing with it so the values were not been linked to the bubble sort.

```
int StSortQ()
{

char sref[10][10];
int intquant[10];
char tempquant[4];
int count;
int n;

ReadBackStockFile();
BubbleSort(sref,intquant,nsti);
cout<<"\nSorted Stock Quantities: \n";
PrintArray(sref, intquant, nsti);
```

To fix this I used a for loop which ran through the file to retrieve the stock references and the quantities to convert into variables to be used in the sort.

```
ReadBackStockFile();
for(count=0;count<nsti;count++)
{
strcpy(tempquant,quantity[count]);
sscanf(&tempquant[0],"%d",&intquant[count]);
strcpy(sref[count],stockref[count]);
getch();
} //end for
BubbleSort(sref,intquant,nsti);
cout<<"\nSorted Stock Quantities: \n";
PrintArray(sref, intquant, nsti);
```



```
if(intvol<60)
{
    valid = 1;
} // end if
```

```
if(intvol>5 && intvol <60)
{
    valid = 1;
} // end if
```

```
if(len!=0)
{
    if(intvol>5 && intvol <60)
    {
        valid = 1;
    } // end if
}
```

When doing range checks I found multiple logic errors.

The first one being due to the range I allowed it allowed the user to enter a volume of 0 for a tin of paint. This is not very realistic and a user is more likely to have missed the first digit so to stop the mistake from being saved I changed the parameters of range ensuring it has to be bigger than 5 to be accepted.

Furthermore, if the user was to accidentally press enter and skip the input it moved on and saved the volume as empty. This would obviously not be good since the user will always have a volume so to stop this I introduced a length check on the value entered first so that if the length was 0 it makes the user try again and re-enter something.

I then went and changed all range checks to ensure this didn't happen anywhere else.

```
cout<<"\nEnter stock reference: ";  
cin.getline(stokref[nsti],3);
```

I found an error that it wasn't allowing me to run the program to check if the add stock routine was working. After following the error message that was given, I realised I had mistyped my variable.

Once I fixed this mistake it allowed the program to run, and stock was added correctly.

```
cout<<"\nEnter stock reference: ";  
cin.getline(stockref[nsti],3);
```

```

Change stock price
*****
Enter the stock reference of the stock you want to change: 1

Current quantity: 5
Colour: White
Type: Matt
Stock Price: 55

Is this the correct item of stock to be amended (0=Yes, 1=No): 0

```

```

compare = strcmpi(looking, stockref[find]); //compares the stock references in the fi
if(compare ==0)                               //If a match is found, checks with the user that
{
    cout<<"\nCurrent quantity: "<<quantity[find];
    cout<<"\nColour: "<<colour[find];
    cout<<"\nType: "<<type[find];
    cout<<"\nStock Price: "<<stockprice[find];
    cin.get();
    cout<<"\nIs this the correct item of stock to be amended (0=Yes, 1=No): ";
    cin>>result;                               //If user enters 1, re
    if(result == 0)                             //If user enters 0, pri
    {
        cin.get();
        while(price == 0)
        {
            cout<<"\nEnter stock price: ";

```

```

cin>>result;
if(result == 0)
{

```

When testing the change stock price function, I found it wasn't allowing me to actually change anything as the next line to enter the new price was not outputting to screen but in fact the function closed and returned to the stock menu.

Since I could see the information about the item of stock I knew there was no readback error so had to be to do with the confirmation of the item of stock.

After looking at the code I saw that I had not checked the value of the result properly.

I had used a singular '=' instead of two as seen in the screenshot.

Once I changed this the output appeared and allowed me to enter the new price which was then saved to the file

```

1
for (del=find;del<nsi;del++)      //If user enters 0, .
{
    strcpy(stockref[del], stockref[del+1]);
    strcpy(quantity[del], quantity[del+1]);
    strcpy(colour[del], colour[del+1]);
    strcpy(volume[del], volume[del+1]);
    strcpy(type[del], type[del+1]);
    strcpy(stockprice[del], stockprice[del+1]);
} // end for
} //end if
else

```

When deleting an item of stock I found that in the for loop it was looping through the number of staff members rather than the number of items of stock.

```

.
for (del=find;del<nsti;del++)      //If user enters 0.
{
    strcpy(stockref[del], stockref[del+1]);
    strcpy(quantity[del], quantity[del+1]);
    strcpy(colour[del], colour[del+1]);
    strcpy(volume[del], volume[del+1]);
    strcpy(type[del], type[del+1]);
    strcpy(stockprice[del], stockprice[del+1]);
} // end for
// ...

```

Since nsi = number of staff as integer rather than nsti. I changed this to the correct global variable

```
View Stock by Stock Reference
*****
Enter stock reference: 1

Colour: White
Stock Price: 55
Quantity: 5
Type: Matt
Volume: 15
Stock reference has not been found.
```

```
//local variables
char looking[25];
char compare;
int find;
```

```
.
//local variables
char looking[25];
int compare;
int find;
```

When trying to view an item of stock the function worked correctly and showed the stock information however it was still producing an error message.

After looking back at the code I saw that I had declared the local variable 'compare' as char rather than int.

Once this had been amended the view function worked correctly.

Schedule

```
H:\Y12\Computer Science\Coursework\4 - Prototype\Prototype.exe
in readback
View Booking
Staff Reference: 1
07 08 09 10 11 12 13 14 15 16 17 18 19
Mon * * * * *
Tue * * * * *
Wed * * * * *
Thu * * * * *
Fri * * * * *
Sat * * * * *
Sun * * * * *
```

Clear schedule before any bookings

```
H:\Y12\Computer Science\Coursework\4 - Prototype\Prototype.exe
Add Booking
*****
in readback
Enter quote number: 5
Enter staff member: 1
Enter date (1-7, Mon-Sun): 3
Enter hour (7-19): 10
```

Booking to be added

```
in readback
View Booking
Staff Reference: 1
07 08 09 10 11 12 13 14 15 16 17 18 19
Mon * * * * *
Tue * * * * *
Wed * * * 5 * * * * *
Thu * * * * *
Fri * * * * *
Sat * * * * *
Sun * * * * *
```

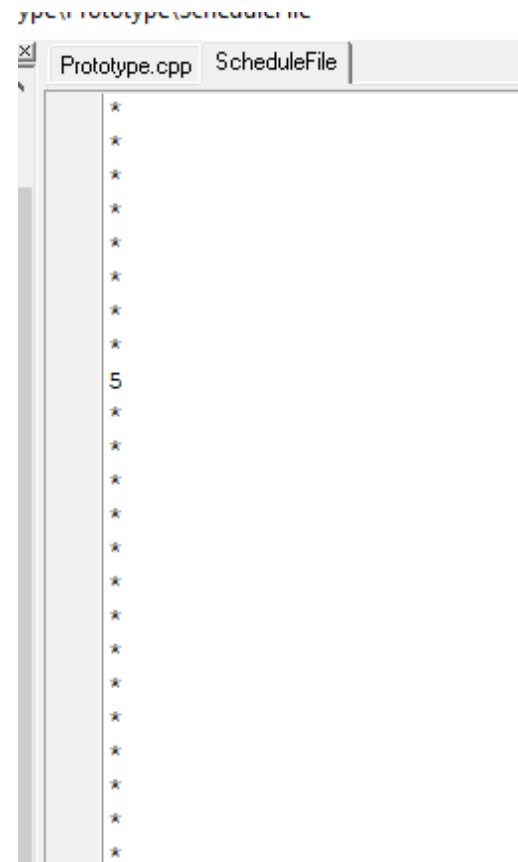
Shows booking added onto the schedule before program is closed however...

```
in readback
View Booking
Staff Reference: 1
    07    08    09    10    11    12    13    14    15    16    17    18    19
Mon    *
Tue
Wed
Thu
Fri
Sat
Sun
```

...when you close the program and re run the data is no longer saved.

This suggested when you added a booking it was not rewriting to the file correctly meaning there was something wrong with my rewrite function.

However I knew this could not be the case since in the actual file I could see where the quote ref had been saved meaning it was writing correctly.



Since the rewrite was working correctly it meant there was an issue with the readback.

I used a dummy print in the readback function to see if it was correctly reading back the date and hour.

```
int ReadBackScheduleFile()
{
    cout<<"\n in readback";
    getch();

    ifstream fin(fileName7, ios::in);
    for(staff=0;staff<2;staff++)
    {
        for(date=0;date<7;date++)
        {
            for(hour=0;hour<13;hour++)
            {
                fin.getline((char*)&booking[staff][date][hour],3);
                cout<<"\n"<<date<<hour<< " booking[date][hour];
            }//end for date
        }//endfor hour
    }//endfor - staff
    getch();
    fin.close();
    return 0;
}
```

This then showed that it was only reading back the first position of the file. Therefore, this meant there was an issue with the length of the variables and not being able to correctly read them back.

H:\Y12\Computer Science\Cou

```
in readback
00*
01
02
03
04
05
06
07
08
09
010
```

To solve this I used trial and error changing the length until I found that the combination in the image below worked and it successfully read back the file.

```
int ReadBackScheduleFile()
{
    cout<<"\n in readback";
    getch();

    ifstream fin(fileName7, ios::in);
    for(staff=0;staff<2;staff++)
    {
        for(date=0;date<7;date++)
        {
            for(hour=0;hour<13;hour++)
            {
                fin.getline((char*)&booking[staff][date][hour],3);
                cout<<"\n"<<date<<hour<< " booking[staff][date][hour];
            }//end for date
        }//endfor hour
    }//endfor - staff
    getch();
    fin.close();
    return 0;
}

int ReWriteScheduleFile()
{
    cout<<"\n in rewrite";
    ofstream fout(fileName7, ios::binary);
    for(staff=0;staff<2;staff++)
    {
        for(date=0;date<7;date++)
        {
            for(hour=0;hour<13;hour++)
            {
                fout.write((char*)&booking[staff][date][hour],1);
                fout.write("\n",1);
            }//endfor
        }//endfor
    }// endfor
    getch();
    fout.close();
    return 0;
}
```

```

ReadBackScheduleFile();
for(staff=0;staff<2;staff++)
{
    cout<<"\n Staff Reference: "<<staff +1;
    for(date=value;date<=value;date++)
    {
        for(hour=0;hour<13;hour++)
        {
            cout<<"\n Quote reference: "<<quoteno;
            LocateQuoteWork(quoteno);
        }//endfor - hour
    }//endfor - date
}// endfor - staff

getch();
return 0;
}

```

The problem I faced was that for the 'Todays Work' Function it was repeating the quote reference multiple times.

To fix this, the date I was searching for was fixed so there was no need to include a for loop for the date so I deleted this.

Then, since I was printing the hours, this was the reason the quote was repeating so many times so to help this I printed all the hours so it made sense on the screen.

However the quote reference which was stored on the schedule was stored under the variable 'booking[staff][date][hour]' so I needed to make sure I was passing to find the quote rather than the variable quoteno.

Once these changes were made the function worked correctly.

```

ReadBackScheduleFile();
for(staff=0;staff<2;staff++)
{
    cout<<"\n Staff Reference: "<<staff +1;
    for(hour=0;hour<13;hour++)
    {
        cout<<"\n"<<hour+7<<"\t";
        cout<<booking[staff][date][hour];
        LocateQuoteWork(booking[staff][date][hour]);
    }//endfor - hour
}// endfor - staff

getch();
return 0;
}

```

```

int Staff1()
{
    ReadBackScheduleFile();
    cout<<"\nView Booking";
    cout<<"\n*****";
    for (staff=0;staff<1;staff++)
    {
        cout<<"\nStaff Reference: "<<staff +1;
        cout<<"\n*****";
        cout<<"\n\t\t07\t08\t09\t10\t11\t12\t13\t14\t15\t16\t17\t18\t19";
        for (date=0;date<366;date++)
        {
            DateOutput;
            for (hour=0;hour<13;hour++)
            {
                cout<<"\t" <<booking[staff][date][hour];
            }//endfor - hour
        }//endfor - date
    }// endfor - staff
    getch();
    return 0;
}

int Staff1()
{
    ReadBackScheduleFile();
    cout<<"\nView Booking";
    cout<<"\n*****";
    for (staff=0;staff<1;staff++)
    {
        cout<<"\nStaff Reference: "<<staff +1;
        cout<<"\n*****";
        cout<<"\n\t\t07\t08\t09\t10\t11\t12\t13\t14\t15\t16\t17\t18\t19";
        for (date=0;date<366;date++)
        {
            DateOutput (date);
            for (hour=0;hour<13;hour++)
            {
                cout<<"\t" <<booking[staff][date][hour];
            }//endfor - hour
        }//endfor - date
    }// endfor - staff
    getch();
    return 0;
}

```

I had an error that the dates outputted for the schedule would not output correctly and would begin to repeat once it reached the end of the year – it would restart.

Furthermore, it wouldn't read back the schedule and would show up as blank this was also because it wasn't taking in the date for the readback.

The issue was that the loop was repeating over again rather than taking the date as a parameter from the loop so I changed the function so that it took the date.

This then meant that it read through and showed the quotes properly and also that it only outputted the year once.

```

return 0;
}

ConvertDate(char datein[12])
{
    int year;
    int compyr;
    int month;
    int day;
    int value=0;
    int valid=0;
    sscanf(&datein[6],"%d", &year);
    sscanf(&datein[4],"%d", &month);
    sscanf(&datein[0],"%d",

```

```

    if((floor(year/4))==(year/4))
    {
        cout<<floor(year/4);
        cout<<year/4;
        valid=1;
    }
    else
    {
        valid=0;
    }

```

```

    if(month==1)
    {
        value = day;
    }

```

```

ConvertDate(char datein[12])
{
    floor year;
    int month;
    int day;
    int value=0;
    int valid=0;
    sscanf(&datein[6],"%f", &year);
    sscanf(&datein[4],"%d", &month);
    sscanf(&datein[0],"%d", &day);

    if((floor(year/4))==(year/4))
    {
        cout<<"\n floor: "<<floor(year/4);
        cout<<"\n divide 4: "<<year/4;
        valid=1;
    }
    else
    {
        valid=0;
    }

```

Enter date: 12/03/2024

floor: 506
divide 4: 506
IN LEAP
72

Enter date: 12/03/2023

floor: 505
divide 4: 505
IN LEAP
72

I found a problem that all years were showing as a leap year. This was because I declared the variable 'year' as an integer so it was ignoring the decimal so therefore both the divisions were always equal. Once I changed 'year' to a float the division comparison worked correctly and the leap year was only applied to the correct years

Enter date: 12/03/2023

71

Enter date: 12/03/2024

floor: 506
divide 4: 506
IN LEAP
72

When trying to change the date of a job the old booking on the schedule wasn't being completely deleted as shown in the screen shot below.

```
//Deletes old booking from the schedule
while(endhour>hour)
{
    strcpy(booking[staff-1][date-1][hour-7],"*");
    hour = hour+1;
} //end while

//Add amended booking to the schedule
```

29/02/2024	*	*	*	*	*	*	*	*	*	*	*	*	*
1/03/2024	1	1	1	1	1	1	1	1	1	1	*	*	*
2/03/2024	*	*	*	*	*	*	*	*	*	*	*	*	*
1/03/2024	*	*	*	*	*	*	*	*	*	1	*	*	*

I realised this was an error with my while loop as it wasn't including the hour value so once I set it to be also equal to the hour the whole booking was deleted.

```
//Deletes old booking from the schedule
while(endhour>=hour)
{
    strcpy(booking[staff-1][date-1][hour-7],"*");
    hour = hour+1;
} //end while
```

Quotes

I had an error that when the date was being validated to check it was in the future it was allowing the next field to be entered even if the date was in the past.

First I checked that the values were being correctly found from both the systems clock and the user.

This shows that all the values entered were correct so then I checked the validation routine which was also logically correct.

I then checked that the routines were returning the correct value of the variable 'valid' - they were.

Therefore, I went right back to the while loop where it is entered and found a logic error where both validation routines had to come back as incorrect for the date to be re-entered rather than just one. To fix this I changed it from AND to OR.

```
cout<<"\n Systems Clock Values:";
cout<<"\n day: "<<dayst;
cout<<"\n month: "<<monthint;
cout<<"\n year: "<<year;

cout<<"\n User Values:";
cout<<"\n day: "<<day;
cout<<"\n month: "<<month;
cout<<"\n year: "<<year;
```

Enter date quote is produced: 12/02/2022

Systems Clock Values:
day: 14
month: 2
year: 2024
User Values:
day: 12
month: 2
year: 2022

```
while(dateq == 0 && systems ==0)
{
    cout<<"\nEnter date quote is produced: ";
    cin.getline(quotedate[nqi],11);
    dateq = DateVal(quotedate[nqi]);
    systems = SystemsClockVal(quotedate[nqi]);
    cout<<dateq;
    cout<<systems;
} // end while
```

12/02/2022

10

```
while(dateq == 0 && systems ==0)
```

```
while(dateq == 0 || systems ==0)
```



```
while(dateq == 0 || systems ==0)
{
    cout<<"\nEnter date quote is produced: ";
    cin.getline(quotedate[nqi],11);
    dateq = DateVal(quotedate[nqi]);
    systems = SystemsClockVal(quotedate[nqi]);
} // end while
```

Another issue was that when entering the date that the quote was produced the validation I used made it so that the date had to be either today's date or the future however this does not make sense because a quote may have been created in the days before.

To solve this, I removed that validation from the routine and deleted the local variable, this meant that the format of the date must still be correct however it could be in the past.

```
while(dateq == 0)
{
    cout<<"\nEnter date quote is produced: ";
    cin.getline(quotedate[nqi],11);
    dateq = DateVal(quotedate[nqi]);
} // end while
```

When adding a quote, I found an issue that it wasn't retrieving the customer information to view it once it had been saved to the file.

I knew that all the information had been written to the file as I could see it in the customer file as shown below, therefore I knew it was a readback issue.

After looking at my customer file I saw that the customer I had entered to link didn't actually exist. Once I added a new quote with a customer actually stored in the file the customer information was outputted.

```
Add Quote
*****
Enter quote reference: 4

Enter customer reference: 4

Enter date quote is produced: 26/02/2024

Enter job description: Paint Kitchen White

Enter number of days on the job: 5

Enter travel costs: 15

How many items of stock are required: 1

Enter stock reference: 1

Stock Price: 45
What quantity is required: 3

Price Breakdown
*****
Materials: 135
Labour: 750
Mileage: 15
VAT: 720
Total: 1620_
```

DevTestProgram.cpp	QuotesFileSD
	4
	4
	26/02/2024
	Paint Kitchen White
	5
	750
	15
	720
	135
	1620

```
View Quote by Quote Reference
*****
Enter quote reference: 4

Customer reference: 4

Job Description: Paint Kitchen White

Number of days worked: 5

Price Breakdown
*****
Materials:      £135
Labour:         £750
Mileage:        £15
Total Cost:     £1620_
```

```
for(del=find;del<nqi;del++)
{
    strcpy(quoteref[del], quoteref[del+1]);
    strcpy(custno[del], custno[del+1]);
    strcpy(quotedate[del], quotedate[del+1]);
    strcpy(mainjobdesc[del], mainjobdesc[del+1]);
    strcpy(numofday[del], numofdays[del+1]);
    strcpy(labourq[del], labourq[del+1]);
    strcpy(mileage[del], mileage[del+1]);
    strcpy(vat[del], vat[del+1]);
    strcpy(stockcost[del], stockcost[del+1]);
    strcpy(totalcost[del], totalcost[del+1]);
} // end for
} // end if
```

Once I had programmed the DeleteQuote function I tried to run it however it wouldn't let me as I had misspelt a variable.

Once I corrected the spelling mistake the function ran at the quote was successfully deleted.

I tried to view a quote with a reference '1' which I knew was already stored in the quote file however when I entered the reference no information came up.

```
View Quote by Quote Reference
*****
Enter quote reference: 1
```

As I looked back through the code I couldn't see anything wrong with it – logical errors or syntax errors.

I then realised I had not included the read back function for the quotes file so would not find any of the information.

Once I included this, the correct information was outputted

```
int QRef()
{
    //local variables
    char looking[25];
    char custnochar[3];
    int compare;
    int comparecust;
    int find;
    char pound = 156;

    cout<<"\nView Quote by Quote Reference";
    cout<<"\n*****";

    cin.get();

    cout<<"\nEnter quote reference: ";
    cin.getline(looking,25);           //Quote reference to be found
    for(find=0;find<nqi;find++)        //Searches through all the quotes in the file
    {
        compare = strcmpi(looking,quoteref[find]);    //Compares the quote references in th
        if(compare == 0)
        {
            cout<<"\nCustomer reference: "<<custno[find];    //If a match is found, quot
            LocateCust(custno[find]);
            cout<<"\n";
            cout<<"\nJob Description: "<<mainjobdesc[find];
            cout<<"\n";
            cout<<"\nNumber of days worked: "<<numofdays[find];
            cout<<"\n";
            cout<<"\nPrice Breakdown";
            cout<<"\n*****";
        }
    }
}
```

```

for(find=0; find<nqi; find++)           //Sea
{
    compare = strcmpy(looking,quoteref[find]);
    if(compare ==0)
    {
        cout<<"\nCustomer Reference: "<<cus

```

When I tried to use the change quote function the program wouldn't run. This is because I had misspelt the command. Once I re-typed the command the function ran smoothly.

```

{
compare = strcmppi(looking,quoteref[find]);
if(compare ==0)

```

Invoices

When using the system I tried to add an invoice however once I had finished adding in all the fields and left the program it had not been written to the file.

I soon realised this was because I had forgot add one to the number of invoices.

When I added this line of code, the invoice was added correctly to the file.

```
ReWriteInvoiceFile();  
getch();  
return 0;  
}
```

```
ReWriteInvoiceFile();  
nii=nii+1;  
getch();  
return 0;  
}
```

```

cout<<"\nEnter the invoice reference of the invoice you want to change: ";
cin.getline(looking,25);                                //Invoice to be found
for(find =0; find<nii; find++)    //Searches through all the invoices in the invoice file
{
    compare = strcmpi(looking, invoiceref[find]);        //Compares each invoice in the file
    if(compare ==0)
    {
        cout<<"\nCustomer Reference: "<<custnum[nii];
        LocateCustInvoice(custnum[nii]);                //Routine which finds customer
        cout<<"\nQuote Reference: "<<quotenum[nii];
        LocateQuoteCheck(quotenum[nii]);                //Routine which finds quote/box
        cin.get();
        cout<<"\nIs this the correct invoice to be amended (0=Yes, 1=No): ";
        cin>>result;                                    //Checks with user
        if(result == 0)
    }
}

```

When trying to change an invoice the data for the invoice wasn't showing even though I had ensured the invoice reference was correct. When I went back to the program, I realised that I had used the wrong variable when outputting the information. I should have used 'find' as this contained the value of the invoice reference not 'nii' once I made this change it showed the correct information when I entered the invoice reference.

```

cout<<"\nCustomer Reference: "<<custnum[find];
LocateCustInvoice(custnum[find]);
cout<<"\nQuote Reference: "<<quotenum[find];
LocateQuoteCheck(quotenum[find]);
cin.get();

```


When I tried to view the unpaid invoices it showed that the comparison of the status of the payment of invoice and '1' was not working, this was because I declared 1 wrong once I changed it to double speech marks it worked and the unpaid invoices were shown.

```
ReadBackInvoiceFile();
for (find=0; find<nii; find++)           //searches through
{
    compare = strcmpi('1', paid[find]); //compare
    if (compare==0)                      //if found
    {
        cout<<"\nCustomer reference: "<
        LocateCustInvoice(custnum[find]
        // end if
    }
}
```

```
ReadBackInvoiceFile();
for (find=0; find<nii; find++)           //searches through
{
    compare = strcmpi("1", paid[find]); //compare
    if (compare==0)                      //if found
    {
        cout<<"\nCustomer reference: "<
        LocateCustInvoice(custnum[find]
        } // end if
    } // end for
}
```

Staff

```
while(ref==0)
{
    cout<<"\nEnter staff reference: ";
    cin.getline(staffref[nsi],3);
    ref = UniqueStaff(staffref[nsi]);
} // end while
```

```
//local variables
```

```
int insg;
int ref;
int fname;
int lname;
int adone;
int adtwo;
int adthree;
int pcode;
int tel;
int emertel;
int uquser;
int passval;
int levrage;
```

When attempting to add a staff member to the file I discovered it wasn't doing the validation correctly even though the code was correct.

After looking in depth I realised I hadn't declared the local variables correctly for the while loop and needed to set them all equal to zero for it to work.

```
SoftwareDev.cpp
itoa(nsi,nsc,10);

ReWriteStaffFile();
getch();
return 0;
}

int UniqueStaff(char staff[3])
{
int uniqueref=0;
int find;
int compare;

ReadBackStaffFile();
for(find=0;find<nsi;find++)
{
compare = strcmpi(staffref[find], staff);
if (compare == 0)
{
uniqueref = 0;
} //endif
} //endfor
if (uniqueref==0)
{
cout<<"\n Staff reference already in use. Please try again.";
getch() ;
} // end if error
return uniqueref;
}
```

I found that my unique check for a staff reference wasn't working and was saying all the references I entered were already in use – as shown below.

I realised I had declared the local variable 'uniqueref' = 0 rather than = 1. This meant that there was no option to break the loop

After I changed this the validation worked correctly.

```
H:\Y12\Computer Science\Coursework\6 - Software Development\SoftwareDev.exe
Add Staff
*****
Enter staff reference: 1

Staff reference already in use. Please try again.
Enter staff reference: 2

Staff reference already in use. Please try again.
Enter staff reference: 3

Staff reference already in use. Please try again.
Enter staff reference: 7

Staff reference already in use. Please try again.
```

```
SoftwareDev.cpp
itoa(nsi,nsc,10);

ReWriteStaffFile();
getch();
return 0;
}

int UniqueStaff(char staff[3])
{
int uniqueref=1;
int find;
int compare;

ReadBackStaffFile();
for(find=0;find<nsi;find++)
{
compare = strcmpi(staffref[find], staff);
if (compare == 0)
{
uniqueref = 0;
} //endif
} //endfor
if (uniqueref==0)
{
cout<<"\n Staff reference already in use. Please try again.";
getch() ;
} // end if error
return uniqueref;
}
```

```

cout<<"\nChange Home Address";
cout<<"\n*****";

cout<<"\nEnter the staff reference of the staff member's home address you wish to change";
cin.getline(looking,25);
for(find =0; find<nsi; find++)
{
    compare = strcmpi(looking, staffref[find]);
    if(compare ==0)
    {
        cout<<"\nFull name: "<<fnamestaff[find]<<" "lnamestaff[find];
        cout<<"\nMobile Number: "<<telnostaff[find];
        cin.get();
        cout<<"\nIs this the correct member of staff to be amended (Y/N)";
    }
}

```

When trying to use a change function for the staff it wouldn't let me run the program.

This was because I had forgot the chevrons on the output to check it was the correct member of staff.

Once I edited this the function ran correctly.

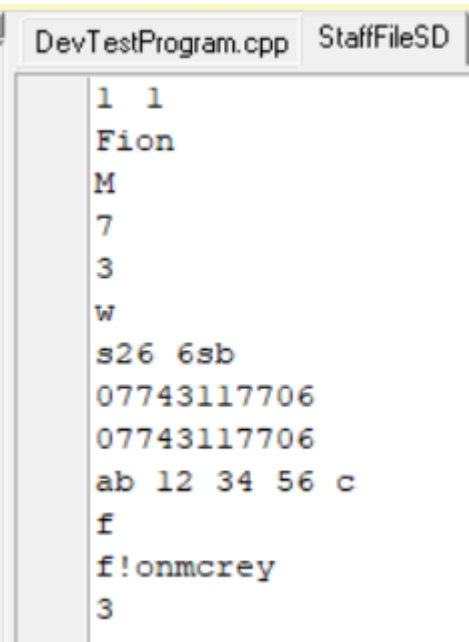
```
View Staff Member by Staff Reference
*****
Enter staff reference: 1

Staff reference not found.
```

When I tried to view a staff member it was producing an error message that the staff reference did not exist.

However, as seen in the screenshot below you can see a staff member with reference 1 stored in file.

After looking back at the code I realised I had use the wrong read back function. Once I changed this to 'ReadBackStaffFile' the view function worked correctly.

A screenshot of a code editor with two tabs: 'DevTestProgram.cpp' and 'StaffFileSD'. The 'StaffFileSD' tab is active and displays a text file containing a staff member record for reference 1. The record is formatted as follows:

```
1 1
Fion
M
7
3
w
s26 6sb
07743117706
07743117706
ab 12 34 56 c
f
f!onmcrey
3
```

```
cout<<"\nView Staff Member by Staff Reference";
cout<<"\n*****";
ReadBackStockFile();
cin.get();
```

Security

An issue I found when attempting to set up the security was that I couldn't get the level of access to work throughout the program to check if they had access or not.

So I had the error that it was allowing anyone to enter all parts of the system to try and overcome this I decided to extract the access level of the staff member as they logged in.

By doing this, I could convert the access level to a global variable to be used throughout the program.

The code for this is shown below.

```
//Finds the level of access for hierarchial use
int FindLoa(char user[15])
{
    //local variables
    int find;
    int compare;
    int comparetwo;
    int valid=0;

    for(find=0;find<nsi;find++)           //Searches t
    {
        compare = strcmpi(user,username[find]);
        if(compare==0)
        {
            level = atoi(loa[find]);
        } //end if
    } //end for
    getch();
    return valid;
}
```