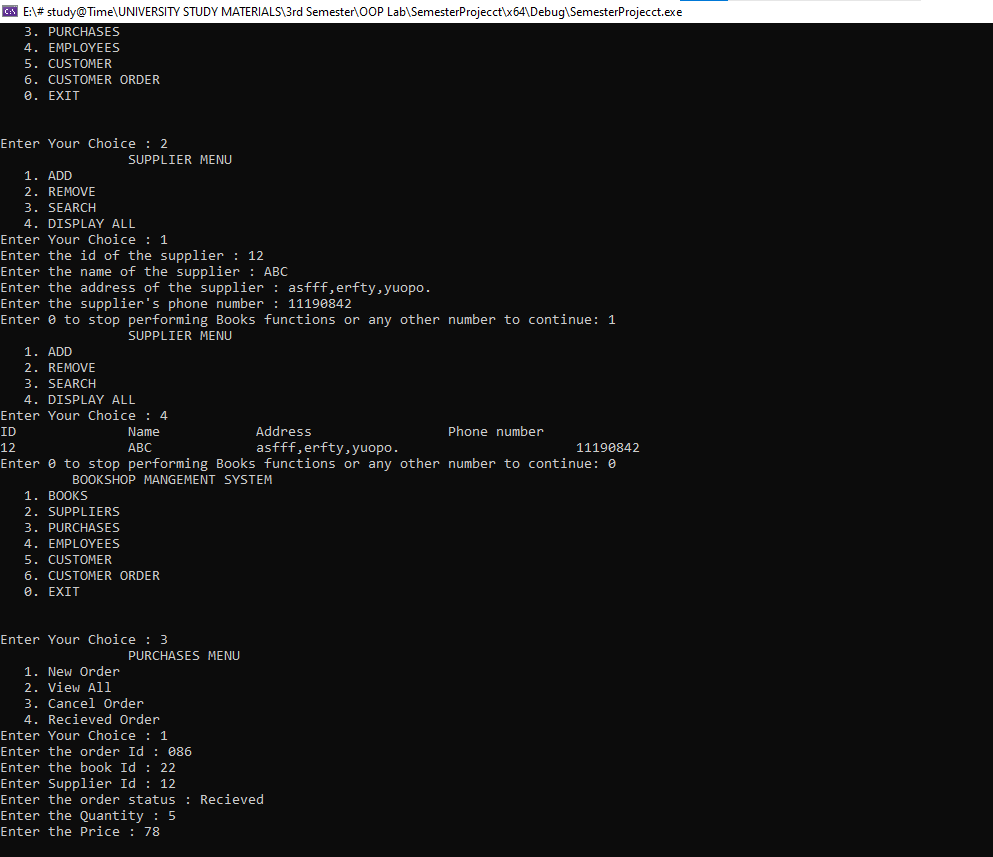
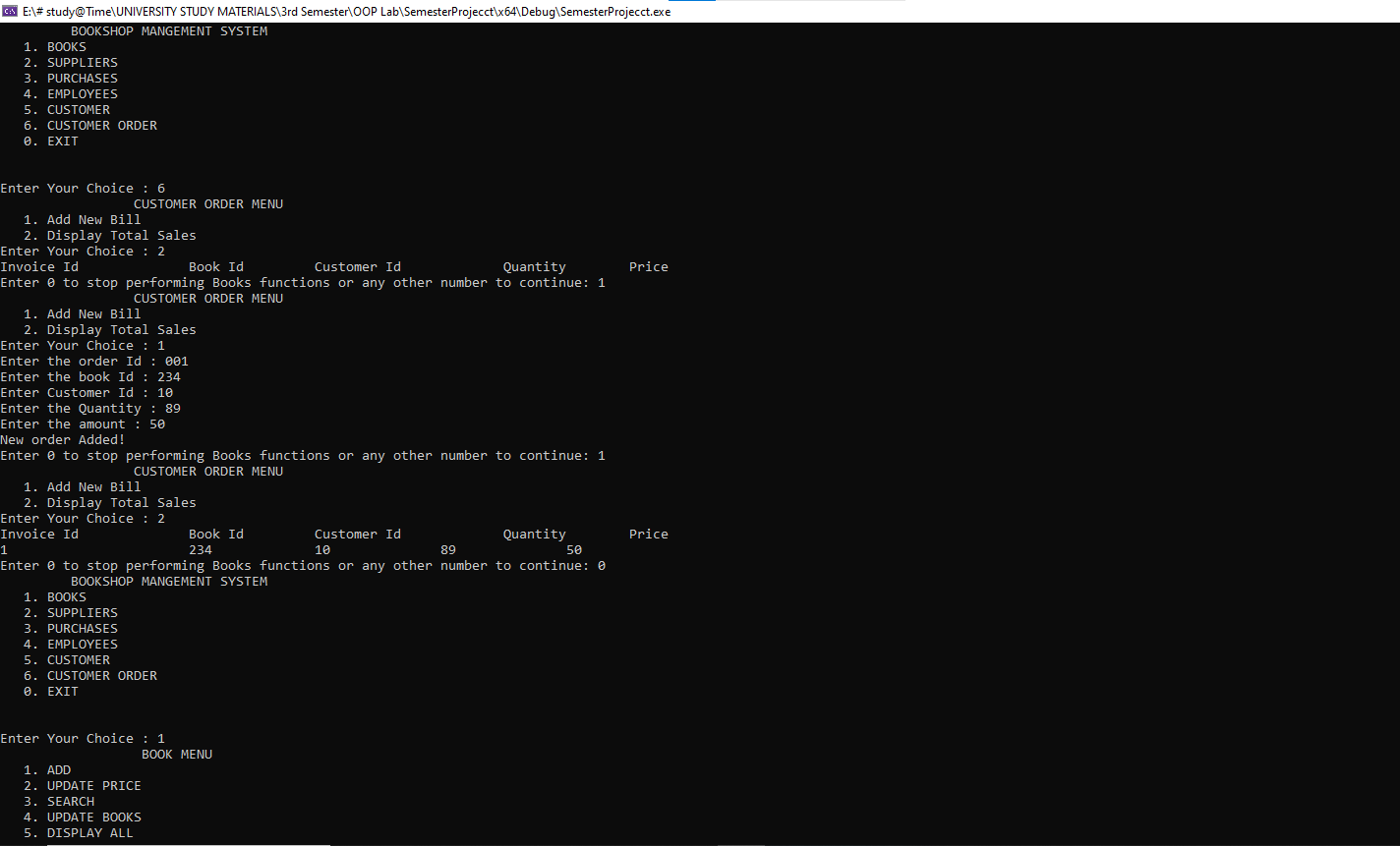


OOP Semester Project S23

**MUSFIRAH ZAIB**

**L1S23BSSE0107**





#include<iostream>

#include<string>

using namespace std;

class Person

{

protected:

int id;

string name;

string phone\_no;

string address;

};

class books

{

private:

int id;

string name;

string author;

int price;

int quantity;

public:

static int no\_of\_books;

public:

books\* add(books\* b);

books\* update\_price(books\* b);

void search(books\* b);

books\* update(books\* b);

void display(books\* b);

books\* book\_menu(books\* b);

};

books\* books::add(books\* b)

{

string n, a;

int o, p, q;

cout << "Enter the id of the book : ";

cin >> o;

cin.ignore();

cout << "Enter the name of the book : ";

getline(cin, n);

cout << "Enter the name of the author : ";

getline(cin, a);

cout << "Enter the Price : ";

cin >> p;

cout << "Enter the quantity Recived : ";

cin >> q;

books\* ptr = new books[no\_of\_books + 1];

int i;

for (i = 0; i < no\_of\_books; i++)

{

\*(ptr + i) = \*(b + i);

}

ptr[i].id = o;

ptr[i].name = n;

ptr[i].author = a;

ptr[i].price = p;

ptr[i].quantity = q;

no\_of\_books++;

delete[]b;

b = nullptr;

return ptr;

}

books\* books::update\_price(books\* b)

{

int p, bid;

cout << "Enter the id of the book whose price you want to update : ";

cin >> bid;

cout << "Enter the updated Price : ";

cin >> p;

books\* ptr = new books[no\_of\_books];

for (int i = 0; i < no\_of\_books; i++)

{

(ptr + i)->id = (b + i)->id;

(ptr + i)->name = (b + i)->name;

(ptr + i)->author = (b + i)->author;

(ptr + i)->quantity = (b + i)->quantity;

if (b[i].id == bid)

{

(ptr + i)->price = p;

}

else

{

(ptr + i)->price = (b + i)->price;

}

}

delete[]b;

b = nullptr;

return ptr;

}

void books::search(books\* b)

{

int n;

cout << "Enter the id of the book who price you want to search : ";

cin >> n;

for (int i = 0; i < no\_of\_books; i++)

{

if (b[i].id == n)

{

cout << "Book id is : " << (b + i)->id << endl;

cout << "Name of book is : " << (b + i)->name << endl;

cout << "Name of Author : " << (b + i)->author << endl;

cout << "Price : " << (b + i)->price << endl;

cout << "Quantity : " << (b + i)->quantity << endl;

}

}

}

books\* books::update(books\* b)

{

int bid, nbid, p, q;

string n, a;

cout << "Enter the id of the book who price you want to update : ";

cin >> bid;

cout << "Enter the id of the book : ";

cin >> nbid;

cin.ignore();

cout << "Enter the name of the updated book : ";

getline(cin, n);

cout << "Enter the name of the author : ";

getline(cin, a);

cout << "Enter the Price : ";

cin >> p;

cout << "Enter the quantity Recived : ";

cin >> q;

books\* ptr = new books[no\_of\_books];

for (int i = 0; i < no\_of\_books; i++)

{

if (b[i].id == bid)

{

ptr[i].id = nbid;

ptr[i].name = n;

ptr[i].author = a;

ptr[i].price = p;

ptr[i].quantity = q;

}

else

{

ptr[i].id = b[i].id;

ptr[i].name = b[i].name;

ptr[i].author = b[i].author;

ptr[i].price = b[i].price;

ptr[i].quantity = b[i].quantity;

}

}

delete[]b;

b = nullptr;

return ptr;

}

void books::display(books\* b)

{

cout << "ID Book Name Author Name Price Quantity" << endl;

for (int i = 0; i < no\_of\_books; i++)

{

cout << (b + i)->id << " " << (b + i)->name << " " << (b + i)->author << " " << (b + i)->price << " " << (b + i)->quantity << endl;

}

}

books\* books::book\_menu(books\* b)

{

int c, e;

do {

cout << " BOOK MENU" << endl;

cout << " 1. ADD" << endl;

cout << " 2. UPDATE PRICE" << endl;

cout << " 3. SEARCH" << endl;

cout << " 4. UPDATE BOOKS" << endl;

cout << " 5. DISPLAY ALL" << endl;

cout << "Enter Your Choice : ";

cin >> c;

if (c == 1)

{

b = add(b);

}

else if (c == 2)

{

b = update\_price(b);

}

else if (c == 3)

{

search(b);

}

else if (c == 4)

{

b = update(b);

}

else if (c == 5)

{

display(b);

}

else

{

cout << "Invalid input" << endl;

}

cout << "Enter 0 to stop performing Books functions or any other number to continue: ";

cin >> e;

} while (e != 0);

return b;

}

class suppliers : public Person

{

public:

static int no\_of\_supplier;

public:

suppliers\* add(suppliers\* s);

suppliers\* remove\_supplier(suppliers\* s);

void search\_id(suppliers\* s);

void display(suppliers\* b);

suppliers\* supplier\_menu(suppliers\* s);

};

suppliers\* suppliers::add(suppliers\* s)

{

suppliers\* b1 = new suppliers;

cout << "Enter the id of the supplier : ";

cin >> b1->id;

cin.ignore();

cout << "Enter the name of the supplier : ";

getline(cin, b1->name);

cout << "Enter the address of the supplier : ";

getline(cin, b1->address);

cout << "Enter the supplier's phone number : ";

cin >> b1->phone\_no;

suppliers\* ptr = new suppliers[no\_of\_supplier + 1];

int i;

for (i = 0; i < no\_of\_supplier; i++)

{

\*(ptr + i) = \*(s + i);

}

(ptr + i)->id = b1->id;

(ptr + i)->name = b1->name;

(ptr + i)->phone\_no = b1->phone\_no;

(ptr + i)->address = b1->address;

no\_of\_supplier++;

delete[]s;

s = nullptr;

delete b1;

b1 = nullptr;

return ptr;

}

suppliers\* suppliers::remove\_supplier(suppliers\* s)

{

int j;

cout << "Enter the id of the supplier you want to remove : ";

cin >> j;

suppliers\* ptr = new suppliers[no\_of\_supplier - 1];

for (int i = 0, k = 0; i < no\_of\_supplier; i++)

{

if (s[i].id != j)

{

\*(ptr + k) = \*(s + i);

k++;

}

}

no\_of\_supplier--;

delete[]s;

s = nullptr;

return ptr;

}

void suppliers::search\_id(suppliers\* s)

{

int j;

cout << "Enter the id of the supplier you want to search : ";

cin >> j;

for (int i = 0; i < no\_of\_supplier; i++)

{

if (s[i].id == j)

{

cout << "Supplier id is : " << (s + i)->id << endl;

cout << "Name of supplier is : " << (s + i)->name << endl;

cout << "Address of supplier : " << (s + i)->address << endl;

cout << "Phone numbers of supplier is : " << (s + i)->phone\_no << endl;

}

}

}

void suppliers::display(suppliers\* b)

{

cout << "ID Name Address Phone number" << endl;

for (int i = 0; i < no\_of\_supplier; i++)

{

cout << (b + i)->id << " " << (b + i)->name << " " << (b + i)->address << " " << (b + i)->phone\_no << endl;

}

}

suppliers\* suppliers::supplier\_menu(suppliers\* s)

{

int c, e;

do {

cout << " SUPPLIER MENU" << endl;

cout << " 1. ADD" << endl;

cout << " 2. REMOVE" << endl;

cout << " 3. SEARCH" << endl;

cout << " 4. DISPLAY ALL" << endl;

cout << "Enter Your Choice : ";

cin >> c;

if (c == 1)

{

s = add(s);

}

else if (c == 2)

{

s = remove\_supplier(s);

}

else if (c == 3)

{

search\_id(s);

}

else if (c == 4)

{

display(s);

}

else

{

cout << "Invalid input" << endl;

}

cout << "Enter 0 to stop performing Books functions or any other number to continue: ";

cin >> e;

} while (e != 0);

return s;

}

class purchases\_Book

{

private:

int order\_id;

int book\_id;

int sup\_id;

int quantity;

int price\_each;

public:

static int no\_of\_purchased\_books;

purchases\_Book\* new\_ord(purchases\_Book\* b);

void view(purchases\_Book\* p);

purchases\_Book\* remove\_order(purchases\_Book\* s);

purchases\_Book\* purchaseBook\_menu(purchases\_Book\* p);

};

purchases\_Book\* purchases\_Book::remove\_order(purchases\_Book\* s)

{

int j;

cout << "Enter the id of the Order you want to remove : ";

cin >> j;

purchases\_Book\* ptr = new purchases\_Book[no\_of\_purchased\_books - 1];

for (int i = 0, k = 0; i < no\_of\_purchased\_books; i++)

{

if (s[i].order\_id != j)

{

\*(ptr + k) = \*(s + i);

k++;

}

}

no\_of\_purchased\_books--;

delete[]s;

s = nullptr;

return ptr;

}

purchases\_Book\* purchases\_Book::new\_ord(purchases\_Book\* p)

{

purchases\_Book b1; //= new purchases\_Book;

cout << "Enter the order Id : ";

cin >> b1.order\_id;

cout << "Enter the book Id : ";

cin >> b1.book\_id;

cout << "Enter Supplier Id : ";

cin >> b1.sup\_id;

cout << "Enter the Quantity : ";

cin >> b1.quantity;

cout << "Enter the Price : ";

cin >> b1.price\_each;

purchases\_Book\* ptr = new purchases\_Book[no\_of\_purchased\_books + 1];

int i;

for (i = 0; i < no\_of\_purchased\_books; i++)

{

\*(ptr + i) = \*(p + i);

}

ptr[i].order\_id = b1.order\_id;

ptr[i].book\_id = b1.book\_id;

ptr[i].sup\_id = b1.sup\_id;

ptr[i].quantity = b1.quantity;

ptr[i].price\_each = b1.price\_each;

no\_of\_purchased\_books++;

cout << "New order Added!\n";

return ptr;

}

void purchases\_Book::view(purchases\_Book\* p)

{

cout << "Order Id Book Id Supplier Id Quantity Price " << endl;

for (int i = 0; i < no\_of\_purchased\_books; i++)

{

cout << p[i].order\_id << " " << p[i].book\_id << " " << p[i].sup\_id << " " << p[i].quantity << " " << p[i].price\_each << endl;

}

}

purchases\_Book\* purchases\_Book::purchaseBook\_menu(purchases\_Book\* p)

{

int c, e;

do {

cout << " PURCHASES MENU" << endl;

cout << " 1. New Order" << endl;

cout << " 2. View All" << endl;

cout << " 3. Cancel Order" << endl;

cout << "Enter Your Choice : ";

cin >> c;

if (c == 1)

{

p = new\_ord(p);

}

else if (c == 2)

{

view(p);

}

else if (c == 3)

{

p = remove\_order(p);

}

else

{

cout << "Invalid input" << endl;

}

cout << "Enter 0 to stop performing Books functions or any other number to continue: ";

cin >> e;

} while (e != 0);

return p;

}

class employees : public Person

{

private:

long int salary;

public:

static int no\_of\_employee;

public:

employees\* add(employees\* b);

void search\_emp(employees\* b);

void display(employees\* b);

employees\* update\_salary(employees\* b);

employees\* employee\_menu(employees\* p);

};

employees\* employees::add(employees\* b)

{

employees\* b1 = new employees;

cout << "Enter the id of the employees : ";

cin >> b1->id;

cin.ignore();

cout << "Enter the name of the employees : ";

getline(cin, b1->name);

cout << "Enter the salary of the employees : ";

cin >> b1->salary;

cout << "Enter the employees's phone number : ";

cin >> b1->phone\_no;

cin.ignore();

cout << "Enter the address of the employee : ";

getline(cin, b1->address);

employees\* ptr = new employees[no\_of\_employee + 1];

int i;

for (i = 0; i < no\_of\_employee; i++)

{

\*(ptr + i) = \*(b + i);

}

(ptr + i)->id = b1->id;

(ptr + i)->name = b1->name;

(ptr + i)->salary = b1->salary;

(ptr + i)->phone\_no = b1->phone\_no;

(ptr + i)->address = b1->address;

no\_of\_employee++;

delete[]b;

b = nullptr;

delete b1;

b1 = nullptr;

return ptr;

}

employees\* employees::update\_salary(employees\* b)

{

int p, eid;

cout << "Enter the id of the employees whose salary you want to update : ";

cin >> eid;

cout << "Enter the updated salary : ";

cin >> p;

employees\* ptr = new employees[no\_of\_employee];

for (int i = 0; i < no\_of\_employee; i++)

{

(ptr + i)->id = (b + i)->id;

(ptr + i)->name = (b + i)->name;

(ptr + i)->phone\_no = (b + i)->phone\_no;

(ptr + i)->address = (b + i)->address;

if (b[i].id == eid)

{

(ptr + i)->salary = p;

}

else

{

(ptr + i)->salary = (b + i)->salary;

}

}

delete[]b;

b = nullptr;

return ptr;

}

void employees::search\_emp(employees\* b)

{

int eid;

cout << "Enter the id of the employees you want to search : ";

cin >> eid;

for (int i = 0; i < no\_of\_employee; i++)

{

if (b[i].id == eid)

{

cout << "Employee id is : " << (b + i)->id << endl;

cout << "Name of employee is : " << (b + i)->name << endl;

cout << "Salary of employee : " << (b + i)->salary << endl;

cout << "Phone number : " << (b + i)->phone\_no << endl;

cout << "Address of employee : " << (b + i)->address << endl;

}

}

}

void employees::display(employees\* b)

{

cout << "ID Employee Name Salary Phone\_number Address" << endl;

for (int i = 0; i < no\_of\_employee; i++)

{

cout << (b + i)->id << " " << (b + i)->name << " " << (b + i)->salary << " " << (b + i)->phone\_no << " " << (b + i)->address << endl;

}

}

employees\* employees::employee\_menu(employees\* p)

{

int c, e;

do {

cout << " EMPLOYEE MENU" << endl;

cout << " 1. New Employee" << endl;

cout << " 2. Search Employee" << endl;

cout << " 3. View All" << endl;

cout << " 4. Update Salary" << endl;

cout << "Enter Your Choice : ";

cin >> c;

if (c == 1)

{

p = add(p);

}

else if (c == 2)

{

search\_emp(p);

}

else if (c == 3)

{

display(p);

}

else if (c == 4)

{

p = update\_salary(p);

}

else

{

cout << "Invalid input" << endl;

}

cout << "Enter 0 to stop performing Books functions or any other number to continue: ";

cin >> e;

} while (e != 0);

return p;

}

class Customer : public Person

{

public:

static int no\_of\_Customers;

public:

Customer\* add(Customer\* b);

void search\_Customers(Customer\* b);

void display(Customer\* b);

Customer\* Customer\_menu(Customer\* p);

};

Customer\* Customer::add(Customer\* b)

{

Customer\* b1 = new Customer;

cout << "Enter the id of the Customer : ";

cin >> b1->id;

cin.ignore();

cout << "Enter the name of the Customer : ";

getline(cin, b1->name);

cout << "Enter the Customer's phone number : ";

cin >> b1->phone\_no;

cin.ignore();

cout << "Enter the address of the Customer : ";

getline(cin, b1->address);

Customer\* ptr = new Customer[no\_of\_Customers + 1];

int i;

for (i = 0; i < no\_of\_Customers; i++)

{

\*(ptr + i) = \*(b + i);

}

(ptr + i)->id = b1->id;

(ptr + i)->name = b1->name;

(ptr + i)->phone\_no = b1->phone\_no;

(ptr + i)->address = b1->address;

no\_of\_Customers++;

delete[]b;

b = nullptr;

delete b1;

b1 = nullptr;

return ptr;

}

void Customer::search\_Customers(Customer\* b)

{

int cid;

cout << "Enter the id of the Customer you want to search : ";

cin >> cid;

for (int i = 0; i < no\_of\_Customers; i++)

{

if (b[i].id == cid)

{

cout << "Customer id is : " << (b + i)->id << endl;

cout << "Name of Customer is : " << (b + i)->name << endl;

cout << "Phone number of Customer is : " << (b + i)->phone\_no << endl;

cout << "Address of Customer : " << (b + i)->address << endl;

}

}

}

void Customer::display(Customer\* b)

{

cout << "ID Customer Name Phone\_number Address" << endl;

for (int i = 0; i < no\_of\_Customers; i++)

{

cout << (b + i)->id << " " << (b + i)->name << " " << (b + i)->phone\_no << " " << (b + i)->address << endl;

}

}

Customer\* Customer::Customer\_menu(Customer\* p)

{

int c, e;

do {

cout << " CUSTOMER MENU" << endl;

cout << " 1. New Customer" << endl;

cout << " 2. Search Customer" << endl;

cout << " 3. Display Customers" << endl;

cout << "Enter Your Choice : ";

cin >> c;

if (c == 1)

{

p = add(p);

}

else if (c == 2)

{

search\_Customers(p);

}

else if (c == 3)

{

display(p);

}

else

{

cout << "Invalid input" << endl;

}

cout << "Enter 0 to stop performing Books functions or any other number to continue: ";

cin >> e;

} while (e != 0);

return p;

}

class Customer\_order

{

private:

int invoice\_id;

int Customer\_id;

int book\_id;

int quantity;

int amount;

public:

static int no\_of\_orders;

public:

Customer\_order\* add(Customer\_order\* p);

void Display\_total\_sales(Customer\_order\* p);

Customer\_order\* CustomerOrder\_menu(Customer\_order\* p);

};

Customer\_order\* Customer\_order::add(Customer\_order\* p)

{

int in, b, c, q, a;

cout << "Enter the order Id : ";

cin >> in;

cout << "Enter the book Id : ";

cin >> b;

cout << "Enter Customer Id : ";

cin >> c;

cout << "Enter the Quantity : ";

cin >> q;

cout << "Enter the amount : ";

cin >> a;

Customer\_order\* ptr = new Customer\_order[no\_of\_orders + 1];

int i;

for (i = 0; i < no\_of\_orders; i++)

{

\*(ptr + i) = \*(p + i);

}

ptr[i].invoice\_id = in;

ptr[i].book\_id = b;

ptr[i].Customer\_id = c;

ptr[i].quantity = q;

ptr[i].amount = a;

no\_of\_orders++;

delete p;

p = nullptr;

cout << "New order Added!\n";

return ptr;

}

void Customer\_order::Display\_total\_sales(Customer\_order\* p)

{

cout << "Invoice Id Book Id Customer Id Quantity Price\n";

for (int i = 0; i < no\_of\_orders; i++)

{

cout << p[i].invoice\_id << " " << p[i].book\_id << " " << p[i].Customer\_id << " " << p[i].quantity << " " << p[i].amount << endl;

}

}

Customer\_order\* Customer\_order::CustomerOrder\_menu(Customer\_order\* p)

{

int c, e;

do {

cout << " CUSTOMER ORDER MENU" << endl;

cout << " 1. Add New Bill" << endl;

cout << " 2. Display Total Sales" << endl;

cout << "Enter Your Choice : ";

cin >> c;

if (c == 1)

{

p = add(p);

}

else if (c == 2)

{

Display\_total\_sales(p);

}

else

{

cout << "Invalid input" << endl;

}

cout << "Enter 0 to stop performing Books functions or any other number to continue: ";

cin >> e;

} while (e != 0);

return p;

}

int books::no\_of\_books = 0, suppliers::no\_of\_supplier = 0, purchases\_Book::no\_of\_purchased\_books = 0;

int employees::no\_of\_employee = 0, Customer::no\_of\_Customers = 0, Customer\_order::no\_of\_orders = 0;

void main\_menu();

int main()

{

int c;

books\* b1 = new books[books::no\_of\_books];

suppliers\* s1 = new suppliers[suppliers::no\_of\_supplier];

purchases\_Book\* p1 = new purchases\_Book[purchases\_Book::no\_of\_purchased\_books];

employees\* e1 = new employees[employees::no\_of\_employee];

Customer\* c1 = new Customer[Customer::no\_of\_Customers];

Customer\_order\* o1 = new Customer\_order[Customer\_order::no\_of\_orders];

do

{

main\_menu();

cin >> c;

if (c == 1)

{

b1 = b1->book\_menu(b1);

}

else if (c == 2)

{

s1 = s1->supplier\_menu(s1);

}

else if (c == 3)

{

p1 = p1->purchaseBook\_menu(p1);

}

else if (c == 4)

{

e1 = e1->employee\_menu(e1);

}

else if (c == 5)

{

c1 = c1->Customer\_menu(c1);

}

else if (c == 6)

{

o1 = o1->CustomerOrder\_menu(o1);

}

else if (c == 0)

{

break;

}

else

{

cout << "Invalid input" << endl;

}

} while (c != 0);

return 0;

}

void main\_menu()

{

cout << " BOOKSHOP MANGEMENT SYSTEM" << endl;

cout << " 1. BOOKS" << endl;

cout << " 2. SUPPLIERS" << endl;

cout << " 3. PURCHASES" << endl;

cout << " 4. EMPLOYEES" << endl;

cout << " 5. CUSTOMER" << endl;

cout << " 6. CUSTOMER ORDER" << endl;

cout << " 0. EXIT" << endl << endl << endl;

cout << "Enter Your Choice : ";

}