This C++ code is a simple console-based inventory management system for a supermarket. It allows the user to create, display, modify, and delete product records, as well as place orders and generate invoices. Below is a breakdown of the code:

### 1. Header Files

```cpp

#include <iostream>

#include <fstream>

#include <cstdio> // for remove and rename functions

#include <cstring>

using namespace std;

```

- \*\*iostream\*\*: Used for standard input/output operations.

- \*\*fstream\*\*: Used for file input/output operations.

- \*\*cstdio\*\*: Used for file operations like remove and rename.

- \*\*cstring\*\*: Used for handling C-style strings.

### 2. Product Class

```cpp

class product

{

int pno;

char name[50];

float price, dis;

public:

void create\_product();

void show\_product() const;

int retpno() const;

float retprice() const;

const char\* retname() const;

int retdis() const;

};

```

- \*\*product\*\*: A class representing a product with attributes such as product number, name, price, and discount.

- \*\*create\_product()\*\*: Method to input product details.

- \*\*show\_product()\*\*: Method to display product details.

- \*\*retpno()\*\*: Returns the product number.

- \*\*retprice()\*\*: Returns the product price.

- \*\*retname()\*\*: Returns the product name.

- \*\*retdis()\*\*: Returns the discount percentage.

### 3. Global Declarations

```cpp

fstream fp;

product pr;

```

- \*\*fp\*\*: File stream object for file operations.

- \*\*pr\*\*: Product object for manipulating product records.

### 4. Writing a Product to File

```cpp

void write\_product()

{

fp.open("Shop.dat", ios::out | ios::app);

pr.create\_product();

fp.write((char\*)&pr, sizeof(product));

fp.close();

cout << "\n\nThe Product Has Been Created ";

}

```

- Opens the file `Shop.dat` in append mode.

- Calls `create\_product()` to input product details.

- Writes the product object to the file.

- Closes the file.

### 5. Displaying All Products

```cpp

void display\_all()

{

cout << "\n\n\n\t\tDISPLAY ALL RECORDS!!!\n\n";

fp.open("Shop.dat", ios::in);

while (fp.read((char\*)&pr, sizeof(product)))

{

pr.show\_product();

cout << "\n\n====================================\n";

}

fp.close();

}

```

- Opens the file `Shop.dat` in input mode.

- Reads and displays all product records from the file.

### 6. Displaying a Specific Product

```cpp

void display\_sp(int n)

{

int flag = 0;

fp.open("Shop.dat", ios::in);

while (fp.read((char\*)&pr, sizeof(product)))

{

if (pr.retpno() == n)

{

pr.show\_product();

flag = 1;

}

}

fp.close();

if (flag == 0)

cout << "\n\nRecord does not exist";

}

```

- Takes a product number `n` as input.

- Searches for the product in the file and displays it if found.

### 7. Modifying a Product

```cpp

void modify\_product()

{

int no, found = 0;

cout << "\n\n\tTo Modify ";

cout << "\n\n\tPlease Enter The Product No. of The Product: ";

cin >> no;

fp.open("Shop.dat", ios::in | ios::out);

while (fp.read((char\*)&pr, sizeof(product)) && found == 0)

{

if (pr.retpno() == no)

{

pr.show\_product();

cout << "\nPlease Enter The New Details of Product" << endl;

pr.create\_product();

int pos = -1 \* sizeof(pr);

fp.seekp(pos, ios::cur);

fp.write((char\*)&pr, sizeof(product));

cout << "\n\n\t Record Updated";

found = 1;

}

}

fp.close();

if (found == 0)

cout << "\n\n Record Not Found ";

}

```

- Takes a product number as input.

- Searches for the product, displays its details, and allows the user to modify it.

- Updates the product details in the file.

### 8. Deleting a Product

```cpp

void delete\_product()

{

int no;

cout << "\n\n\n\tDelete Record";

cout << "\n\nPlease Enter The product no. of The Product You Want To Delete: ";

cin >> no;

fp.open("Shop.dat", ios::in | ios::out);

fstream fp2;

fp2.open("Temp.dat", ios::out);

fp.seekg(0, ios::beg);

while (fp.read((char\*)&pr, sizeof(product)))

{

if (pr.retpno() != no)

{

fp2.write((char\*)&pr, sizeof(product));

}

}

fp2.close();

fp.close();

remove("Shop.dat");

rename("Temp.dat", "Shop.dat");

cout << "\n\n\tRecord Deleted ..";

}

```

- Takes a product number as input.

- Copies all records except the one to be deleted to a temporary file.

- Deletes the original file and renames the temporary file to the original file name.

### 9. Displaying Product Menu

```cpp

void menu()

{

fp.open("Shop.dat", ios::in);

if (!fp)

{

cout << "ERROR!!! FILE COULD NOT BE OPEN\n\n\n Go To Admin Menu to create File";

cout << "\n\n\n Program is closing ....";

exit(0);

}

cout << "\n\n\t\tProduct MENU\n\n";

cout << "====================================================\n";

cout << "P.NO.\t\tNAME\t\tPRICE\n";

cout << "====================================================\n";

while (fp.read((char\*)&pr, sizeof(product)))

{

cout << pr.retpno() << "\t\t" << pr.retname() << "\t\t" << pr.retprice() << endl;

}

fp.close();

}

```

- Displays a list of all products with their numbers, names, and prices.

### 10. Placing an Order and Generating a Bill

```cpp

void place\_order()

{

int order\_arr[50], quan[50], c = 0;

float amt, damt, total = 0;

char ch = 'Y';

menu();

cout << "\n============================";

cout << "\n PLACE YOUR ORDER";

cout << "\n============================\n";

do {

cout << "\n\nEnter The Product No. Of The Product: ";

cin >> order\_arr[c];

cout << "\nQuantity in number: ";

cin >> quan[c];

c++;

cout << "\nDo You Want To Order Another Product? (y/n): ";

cin >> ch;

} while (ch == 'y' || ch == 'Y');

cout << "\n\nThank You For Placing The Order\n";

cout << "\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*INVOICE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "\nPr No.\tPr Name\tQuantity \tPrice \tAmount \tAmount after discount\n";

for (int x = 0; x < c; x++)

{

fp.open("Shop.dat", ios::in);

while (fp.read((char\*)&pr, sizeof(product)))

{

if (pr.retpno() == order\_arr[x])

{

amt = pr.retprice() \* quan[x];

damt = amt - (amt \* pr.retdis() / 100);

cout << "\n" << order\_arr[x] << "\t" << pr.retname() << "\t" << quan[x] << "\t\t" << pr.retprice() << "\t" << amt << "\t\t" << damt;

total += damt;

}

}

fp.close();

}

cout << "\n\n\t\t\t\t\tTOTAL = " << total;

}

```

- Allows the user to place an order by selecting products and their quantities.

- Generates an invoice showing product details, price, amount, discount, and total amount.

### 11. Introduction Function

```cpp

void intro()

{

cout << "SUPER MARKET" << endl;

cout << "BILLING PROJECT" << endl;

cout << "MADE BY : ANUJ KUMAR" << endl;

cout << "SCHOOL : RYAN INTERNATIONAL SCHOOL" << endl;

}

```

- Displays an introductory message with project details.

### 12. Administrator Menu

```cpp

void admin\_menu()

{

char ch2;

cout << "\n\n\n\tADMIN MENU";

cout << "\n\n\t1.CREATE PRODUCT";

cout << "\n\n\t2.DISPLAY ALL PRODUCTS";

cout << "\n\n\t3.QUERY ";

cout << "\n\n\t4.MODIFY PRODUCT";

cout << "\n\n\t5.DELETE PRODUCT";

cout << "\n\n\t6.VIEW PRODUCT MENU";

cout << "\n\n\t7.BACK TO MAIN MENU";

cout << "\n\n\tPlease Enter Your Choice (1-7): ";

ch2 = cin.get();

cin.ignore();

switch (ch2)

{

case '1':

write\_product();

break